

## **General Disclaimer**

### **One or more of the Following Statements may affect this Document**

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.
- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.
- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.
- This document is paginated as submitted by the original source.
- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.



## FUNCTIONAL MODULE PLA2 (PIECEWISE LINEAR ANALYSIS - PHASE 2)

### 4.53 FUNCTIONAL MODULE PLA2 (PIECEWISE LINEAR ANALYSIS - PHASE 2)

#### 4.53.1 Entry Point: PLA2

#### 4.53.2 Purpose

To add the incremental displacement vector, the incremental load vector, and the incremental vector of single-point forces of constraint for the current pass through the Piecewise Linear Analysis Rigid Format DMAP loop to the current running sum of these vectors:

$$\{u_{g_{i+1}}\} = \{u_{g_i}\} + \{\Delta u_{g_i}\}, \quad (1)$$

$$\{P_{g_{i+1}}\} = \{P_{g_i}\} + \{\Delta P_{g_i}\}, \quad (2)$$

$$\{q_{g_{i+1}}\} = \{q_{g_i}\} + \{\Delta q_{g_i}\}. \quad (3)$$

#### 4.53.3 DMAP Calling Sequence

PLA2 DELTAUGV,DELTAPG,DELTAQG/UGV1,PGV1,QG1/V,N,PLACØUNT \$

#### 4.53.4 Input Data Blocks

DELTAUGV - Incremental displacement vector in Piecewise Linear Analysis - g set.

DELTAPG - Incremental load vector in Piecewise Linear Analysis - g set.

DELTAQG - Incremental vector of single-point forces of constraint in Piecewise Linear Analysis - g set.

#### Note:

1. DELTAUGV and DELTAPG cannot be pre-purged.
2. DELTAQG may be pre-purged.

#### 4.53.5 Output Data Blocks

UGV1 - Matrix of successive sums of incremental displacement vectors - g set.

PGV1 - Matrix of successive sums of incremental load vectors - g set.

QG1 - Matrix of successive sums of incremental vectors of single-point forces of constraint - g set.

## MODULE FUNCTIONAL DESCRIPTIONS

### Notes:

1. UGV1 and PGV1 cannot be purged.
2. QG1 may be purged if DELTAQG is purged.

### 4.53.6 Parameters

PLACØUNT - Input and output-integer - this parameter must be set to 1 outside the Piecewise Linear Analysis Rigid Format DMAP loop. This is done using the PARAM module rather than through the Module Properties List (MPL).

### 4.53.7 Method

If PLACØUNT = 1, that is, this is the first time PLA2 has been called in the Piecewise Linear Analysis Rigid Format DMAP loop, then the DELTAUGV data block is copied onto the UGV1 data block. If PLACØUNT > 1, then PLACØUNT is used as a counter to determine how many records (running sum displacement vectors) to skip on the file containing UGV1 so that the most recently computed running sum displacement vector can be read into open core for the vector addition. Once this vector is read into open core, the incremental displacement vector is read and interpreted using subroutines INTPK and ZNTPKI, and the vector addition given in Equation 1 is carried out element-by-element.

Equations 2 and 3 are computed using the method described in the above paragraph.

### 4.53.8 Subroutines

PLA2 has no auxiliary subroutines.

### 4.53.9 Design Requirements

Open core is defined at /PLA2X/.

### 4.53.10 Diagnostic Messages

User message 2127 or 2128 is output if either DELTAUGV (DELTAPG) or UGV1 (PGV1) is purged.



## FUNCTIONAL MODULE PLA3 (PIECEWISE LINEAR ANALYSIS - PHASE 3)

### 4.54 FUNCTIONAL MODULE PLA3 (PIECEWISE LINEAR ANALYSIS - PHASE 3)

#### 4.54.1 Entry Point: PLA3

#### 4.54.2 Purpose

To compute element stresses for nonlinear elements (see definition of linear elements in section 4.52.2) for which the user has requested stress output. It also updates the ESTNL data block so that the output data block, ESTNL1, contains up-to-date element stress information.

#### 4.54.3 DMAP Calling Sequence

PLA3    CSTM,MPT,DIT,DELTAUGV,ESTNL,CASECC/ØNLES,ESTNL1/V,N,PLACØUNT/V,N,PLSETNØ \$

#### 4.54.4 Input Data Blocks

CSTM        - Coordinate System Transformation Matrices.  
MPT        - Material Properties Table.  
DIT        - Direct Input Tables.  
DELTAUGV   - Current incremental displacement vector.  
ESTNL      - Element Summary Table for Nonlinear Elements.  
CASECC     - Case Control Data Table.

#### Notes:

1. CSTM can be purged. However, if some grid point of the model is not in basic coordinates and the CSTM is purged, a fatal error occurs.
2. A fatal error occurs if either MPT, DIT, DELTAUGV, ESTNL or CASECC is purged.

#### 4.54.5 Output Data Blocks

ØNLES      - Nonlinear element stresses (to be processed by the Output File Processor).  
ESTNL1     - Element Summary Table for Nonlinear Elements - Updated.

Note: Neither output data block may be purged.

#### 4.54.6 Parameters

PLACØUNT   - Input-integer-no default value. This is the Piecewise Linear Analysis (PLA)

## MODULE FUNCTIONAL DESCRIPTIONS

Rigid Format DMAP loop counter. It is used in this routine to find the proper loading factors on the PLFACT bulk data card specified by the user (see PLSETNØ below).

PLTSETNØ - Input-integer-no default value. PLSETNØ is the set identification number of some PLFACT bulk data card chosen by the user in his Case Control Deck. It is used to find this PLFACT card in the MPT data block.

### 4.54.7 Method

The module driver, PLA3, is a short routine whose only function is to call subroutines PLA31 and PLA32 which accomplish phase 1 and phase 2 of the task of the module respectively. Subroutine PLA31 reads the incremental displacement vector into core and appends to each element entry of the ESTNL data block the components of the incremental displacement vector corresponding to the grid points of each element. This merged information is written on the scratch data block ESTNLS, GINØ file number 301. In PLA32, the ESTNLS data block is read, and the proper element routine is called to compute element stresses which are prepared in ØFP (Output File Processor) format. Each element routine also updates incremental stress data. The ESTNL data for each element with the updated stress information (but without the components of the displacement vector) are written on ESTNL1.

In PLA31, for TRMEM and QDMEM elements, only the three translational components of the displacement vector at each grid point of the element are appended to the ESTNL entry. Other elements for which Piecewise Linear Analysis is defined use all six components at each grid point.

In PLA32, the difference quotients  $\gamma^*$  and  $\gamma$ , which are the previous and current (with respect to the DMAP loop in the PLA Rigid Format) load increment ratios, are computed as follows. Let  $P_1, P_2, P_3, \dots$ , be the loading factors on a PLFACT bulk data card. Define  $P_0 = 0$ . Define

$$\alpha_i = P_i - P_{i-1} , \quad (1)$$

for  $i \geq 1$ . Then, define  $\gamma_1^* = 0$ , and

$$\gamma_i^* = \frac{\alpha_i}{\alpha_{i-1}} , \quad (2)$$



for  $i > 1$ , and define

$$\gamma_i = \frac{\alpha_{i+1}}{\alpha_i}, \quad (3)$$

for  $i \geq 1$ . These difference quotients are stored in /PLA32C/ for communication to the module's element routines so that they can compute the estimated next strain. The details of the element calculation are given in section 4.87. The input parameter PLACØUNT, being the counter for the PLA Rigid Format DMAP loop, controls the computation of  $\gamma^*$  and  $\gamma$ . However, the module's design assumes (1) PLACØUNT is set to one outside the PLA DMAP loop and (2) module PLA2, which increments PLACØUNT by one, will be executed prior to every DMAP call to PLA3. Hence, the proper choice for the subscript  $i$  in Equations 1, 2 and 3 is one less than the value of PLACØUNT. The difference PLACØUNT-1 is stored in /PLA32C/ as IPASS.

#### 4.54.8 Subroutines

PLA3 uses, for element routine calculations, the utility routines PRETRS, PREMAT, GMMATS and element drivers. Communication of an appended ESTNL element entry to an element routine during phase 2 of PLA3 is accomplished via /PLA32E/, which is 100 words in length. This fact is not explicitly stated below.

The element drivers PSTRM, PSQDM, PSTRI1, PSTRI2, PSQAD1, and PSQAD2, use a) /PLA3ES/, which is 300 words in length, as a communication link for the element subroutines which they call; and b) /PLA3UV/, which is 25 words in length, as a communication link for displacement vectors between the driver and their subroutines. PLA32 will call the element drivers listed above (plus PSRØD and PSBAR); the other subroutines described below (in sections 4.54.8.11 through 4.54.8.18) are only used (directly or indirectly) by the element drivers.

##### 4.54.8.1 Subroutine Name: PLA31

1. Entry Point: PLA31
2. Purpose: To perform phase 1 of the module's operations as described above.
3. Calling Sequence: CALL PLA31

## MODULE FUNCTIONAL DESCRIPTIONS

### 4.54.8.2 Subroutine Name: PLA32

1. Entry Point: PLA32
2. Purpose: To perform phase 2 of the module's operation as described above.
3. Calling Sequence: CALL PLA32

### 4.54.8.3 Subroutine Name: PSRØD

1. Entry Point: PSRØD
2. Purpose: To compute element stresses and to update the ESTNL entry for a RØD, CØNRØD or TUBE element. Note that for a TUBE element, the ESTNL entry is rearranged and elementary transformations are performed in PLA32 so that the PSRØD routine may compute element stresses for a TUBE.
3. Calling Sequence: CALL PSRØD

### 4.54.8.4 Subroutine Name: PSBAR

1. Entry Point: PSBAR
2. Purpose: To compute element stresses and to update the ESTNL entry for a BAR element.
3. Calling Sequence: CALL PSBAR

### 4.54.8.5 Subroutine Name: PSTRM

1. Entry Point: PSTRM
2. Purpose: To calculate the material properties matrix, arrange the flow of element stress calculations and update the ESTNL entry for the TRMEM element.
3. Calling Sequence: CALL PSTRM

### 4.54.8.6 Subroutine Name: PSQDM

1. Entry Point: PSQDM
2. Purpose: To calculate the material properties matrix, arrange the flow of element stress calculations and update the ESTNL entry for the QDMEM element.
3. Calling Sequence: CALL PSQDM



4.54.8.7 Subroutine Name: PSTRI1

1. Entry Point: PSTRI1
2. Purpose: To calculate the material properties matrix, arrange the flow of element stress calculations and update the ESTNL entry for the TRIA1 element.
3. Calling Sequence: CALL PSTRI1

4.54.8.8 Subroutine Name: PSTRI2

1. Entry Point: PSTRI2
2. Purpose: To calculate the material properties matrix, arrange the flow of element stress calculations and update the ESTNL entry for the TRIA2 element.
3. Calling Sequence: CALL PSTRI2

4.54.8.9 Subroutine Name: PSQAD1

1. Entry Point: PSQAD1
2. Purpose: To calculate the material properties matrix, arrange the flow of element stress calculations and update the ESTNL entry for the QUAD1 element.
3. Calling Sequence: CALL PSQAD1

4.54.8.10 Subroutine Name: PSQAD2

1. Entry Point: PSQAD2
2. Purpose: To calculate the material properties matrix, arrange the flow of element stress calculations and update the ESTNL entry for the QUAD2 element.
3. Calling Sequence: CALL PSQAD2

4.54.8.11 Subroutine Name: PSTRM1

1. Entry Point: PSTRM1
2. Purpose: To generate element stress matrices for the TRMEM element, and the membrane portion of TRIA1 and TRIA2 elements, and perform subcomputations for the PSQDM1 routine.
3. Calling Sequence: CALL PSTRM1 (NTYPE)

## MODULE FUNCTIONAL DESCRIPTIONS

NTYPE  $\left\{ \begin{array}{l} 0 = \text{TRMEM, TRIA1, or TRIA2} \\ 1 = \text{Subcomputations for the PSQDM1 subroutine} \end{array} \right.$

### 4.54.8.12 Subroutine Name: PSQDM1

1. Entry Point: PSQDM1
2. Purpose: To generate element stress matrices for the QDMEM element and the membrane portions of QUAD1 and QUAD2 elements.
3. Calling Sequence: CALL PSQDM1

### 4.54.8.13 Subroutine Name: PSTQ1

1. Entry Point: PSTQ1
2. Purpose: To generate element stress matrices for the TRIA1, TRIA2, QUAD1, and QUAD2 elements.
3. Calling Sequence: CALL PSTQ1 (NTYPE)

NTYPE  $\left\{ \begin{array}{l} 1 = \text{TRIA1} \\ 2 = \text{TRIA2} \\ 3 = \text{QUAD1} \\ 4 = \text{QUAD2} \end{array} \right.$

### 4.54.8.14 Subroutine Name: PSTRB1

1. Entry Point: PSTRB1
2. Purpose: To generate element stress matrices for subcalculations of basic bending triangles for the plate portion of TRIA1, TRIA2, QUAD1 and QUAD2 elements.
3. Calling Sequence: CALL PSTRB1 (IØPT)

IØPT  $\left\{ \begin{array}{l} 1 = \text{Subcalculations for PSQPL1} \\ 2 = \text{Subcalculations for PSTPL1} \end{array} \right.$

### 4.54.8.15 Subroutine Name: PSTPL1

1. Entry Point: PSTPL1
2. Purpose: To generate the element stress matrices for the plate portion of TRIA1 and



TRIA2 elements.

3. Calling Sequence: CALL PSTPL1

4.54.8.16 Subroutine Name: PSQPL1

1. Entry Point: PSQPL1

2. Purpose: To generate element stress matrices for the QUAD1 and QUAD2 elements.

3. Calling Sequence: PSQPL1

4.54.8.17 Subroutine Name: PSTRQ2

1. Entry Point; PSTRQ2

2. Purpose: To perform final stress computations for TRMEM and QDMEM elements.

3. Calling Sequence: CALL PSTRQ2 (NTYPE)

$$\text{NTYPE} \begin{cases} 1 = \text{TRMEM element} \\ 2 = \text{QDMEM element} \end{cases}$$

4.54.8.18 Subroutine Name: PSTQ2

1. Entry Point: PSTQ2

2. Purpose: To perform final stress computations for the TRIA1, TRIA2, QUAD1, and QUAD2 elements.

3. Calling Sequence: CALL PSTQ2 (NPTS)

$$\text{NPTS} \begin{cases} 3 = \text{TRIA1 and TRIA2 elements} \\ 4 = \text{QUAD1 and QUAD2 elements} \end{cases}$$

#### 4.54.9 Design Requirements

1. The module was designed so that phase 1 and phase 2 can be executed in separate overlay segments.

2. Open core for phase 1 is defined at /PLA31X/ and for phase 2 at /PLA32X/. Open core requirements for both phases are minimal. In phase 1, the single precision incremental displacement vector in unpacked form must be able to be contained in open core. In phase 2, the CSTM and MPT data blocks, tables in the DIT referenced on MATS1 bulk data cards, and

## MODULE FUNCTIONAL DESCRIPTIONS

the first record (and only record since a PLA problem allows only one CASECC record) of CASECC must be able to be contained in open core.

3. In addition to the common blocks mentioned above, PLA32 uses /PLA32S/, which is 325 words in length, as scratch storage for the module's element routines, and /SOUT/, which is 30 words in length, as a storage buffer for computed element stresses.

4. One scratch file is used, and all arithmetic operations are performed in single precision.

### 4.54.10 Diagnostic Messages

During phase 1, the following diagnostic messages may appear. If the incremental displacement vector is null, user fatal error 3005 will be given. Two system fatal "fail-safe" error messages, 2091 and 2092, may be implemented if the ESTNL input data block was incorrectly constructed in PLA1 or was incorrectly updated during the previous execution of the PLA3 module.

During phase 2, error messages 3001, 3002 or 3003 may occur if the proper loading factors  $P_j$  cannot be found on the PLFACT bulk data card image in the MPT. If the ECPTDS scratch file is not in the prescribed format, system fatal message 2091 will occur.

If the minimal core storage requirements in either phase 1 or phase 2 are not met, the usual fatal error 3008 will occur.



#### 4.55 FUNCTIONAL MODULE PLA4 (PIECEWISE LINEAR ANALYSIS - PHASE 4)

##### 4.55.1 Entry Point: PLA4

##### 4.55.2 Purpose

To generate the stiffness matrix for nonlinear elements,  $[K_{gg}^{nl}]$ , and to update the Element Connection and Properties Table for Nonlinear Elements, ECPTNL, so that it contains up-to-date element stress information.

##### 4.55.3 DMAP Calling Sequence

```
PLA4    CSTM,MPT,ECPTNL,GPCT,DIT,DELTAUGV/KGGNL,ECPTNL1/V,N,PLACØUNT/V,N,PLSETNØ/
        V,N,PLFACT  $
```

##### 4.55.4 Input Data Blocks

CSTM - Coordinate System Transformation Matrices.

MPT - Material Properties Table.

ECPTNL - Element Connection and Properties Table for Nonlinear Elements.

GPCT - Grid Point Connection Table.

DIT - Direct Input Tables.

DELTAUGV - Current incremental displacement vector.

##### Notes:

1. CSTM may be purged. However, if some grid point of the model is not in basic coordinates and the CSTM has been purged, a fatal error will occur.
2. A fatal error occurs if either MPT, ECPTNL, GPCT, DIT or DELTAUGV is purged.

##### 4.55.5 Output Data Blocks

KGGNL - Stiffness matrix of nonlinear elements - g set.

ECPTNL1 - Element Connection and Properties Table for Nonlinear Elements - updated.

Note: Neither KGGNL or ECPTNL1 may be purged.

## MODULE FUNCTIONAL DESCRIPTIONS

### 4.55.6 Parameters

- PLACOUNT - Input-integer-no default value. Loop counter for the Piecewise Linear Analysis (PLA) Rigid Format DMAP loop. The module uses this parameter to find the correct loading factors on the PLFACT bulk data card chosen by the user.
- PLSETNØ - Input-integer-no default value. Set identification number of a PLFACT bulk data card chosen by the user in his Case Control Deck. The module uses this parameter to search the MPT for this card.
- PLFACT - Output-complex-no default value. The difference of loading factors to be used during the next pass of the PLA Rigid Format DMAP loop.

### 4.55.7 Method

The module driver PLA4 is a short routine whose only function is to call subroutines PLA41 and PLA42 which accomplish phase 1 and phase 2 of the task of the module respectively. Subroutine PLA41 reads the incremental displacement vector into core and appends to each element entry of the ECPTNL data block the components of the incremental displacement vector corresponding to the grid points of each element. This merged information is written on the scratch data block ECPTS, GINØ file number 301. In PLA42, the ECPTS data block is processed in a fashion similar to the processing of the ECPT data block in module SMA1 (see the Module Functional Description for SMA1, section 4.27).

In PLA41, for all elements except the BAR element, only the three translational components of the displacement vector at each grid point of an element are appended to the ECPTNL element entry. For a BAR element, all six components of the displacement vector at each grid point are appended.

The logic of the processing of the scratch data block, ECPTS, in PLA42 is very similar to that used in subroutine SMA1A (of SMA1, the stiffness matrix generation module - see the Module Functional Description for SMA1, section 4.27). The similarities are not enumerated here, but notable differences are the following.

1. Before PREMAT is called to read into open core the MPT data block and tables from the DIT data block referenced on MATS1 bulk data cards, the MPT is read in subroutine



PLA42 to compute  $\gamma^*$  and  $\gamma$  as in Equations 1, 2 and 3 in section 4.54, and the real part of the output DMAP parameter PLFACT is set to the value of  $\alpha_{i+1}$  in Equation 3 in section 4.54. The imaginary part of PLFACT is set to zero. The reason for PLFACT being complex is that it is an input parameter to the DMAP module ADD during the next pass of the PLA Rigid Format DMAP loop, and ADD (see section 4.78) requires its parameters to be complex.

2. When PREMAT is called, the last argument is set negative to signal PREMAT that this is a PLA problem and hence that special processing will be required.
3. Subsequent to the call of an element routine, the element type and the updated ECPT entry are written onto the ECPTNL1 data block.

#### 4.55.8 Subroutines

PLA4 uses PRETRD, PRETRS, PREMAT, INVERS, INVERD, GMMATS, and GMMATD as utility routines. The common block /PLA42E/ is the means of communicating a) the element entry of the ECPTS from PLA42 to an element stiffness matrix generation routine and b) the ECPTS element entry with updated stress information from the element routine back to PLA42 upon completion of element matrix generation. This fact is not explicitly stated in the descriptions of the element routines (e.g., PKR0D) given below.

The element drivers PKTRM, PKQDM, PKTRI1, PKTRI2, PKQAD1, and PKQAD2 use a) /PLA4ES/, which is 300 words in length, and b) /PLA4UV/, which is 25 words in length, as communication links with the subroutines that they call. PLA42 will call the drivers listed above which will use (directly and indirectly) the subroutines described below in sections 4.55.8.12 through 4.55.8.22.

##### 4.55.8.1 Subroutine Name: PLA41

1. Entry Point: PLA41
2. Purpose: See discussion above.
3. Calling Sequence: CALL PLA41

##### 4.55.8.2 Subroutine Name: PLA42

1. Entry Point: PLA42
2. Purpose: See discussion above.

## MODULE FUNCTIONAL DESCRIPTIONS

3. Calling Sequence: CALL PLA42

### 4.55.8.3 Subroutine Name: PLA4B

1. Entry Point: PLA4B

2. Purpose: To add a double precision 6 by 6 element stiffness matrix to the "submatrix" corresponding to the current pivot point. This routine performs the same function as, and is modeled after, subroutine SMA1B of module SMA1.

3. Calling Sequence: CALL PLA4B (KE,J)

KE - Row-stored double precision 6 by 6 matrix to be added to the submatrix in core - input.

J - The column index of the KGGNL matrix which corresponds to first column of the KE matrix - integer - input.

### 4.55.8.4 Subroutine Name: PKRØD

1. Entry Point: PKRØD

2. Purpose: To generate the element stiffness matrix for a RØD element and to update the ECPTNL element entry for a RØD element.

3. Calling Sequence: CALL PKRØD

### 4.55.8.5 Subroutine Name: PKBAR

1. Entry Point: PKBAR

2. Purpose: To generate the element stiffness matrix for a BAR element and to update the ECPTNL element entry for a BAR element.

3. Calling Sequence: CALL PKBAR

### 4.55.8.6 Subroutine Name: PKTRM

1. Entry Point: PKTRM

2. Purpose: To calculate the material properties matrix, update the ECPTNL entry, and arrange the flow of element stiffness calculations for the TRMEM element.

3. Calling Sequence: PKTRM



4.55.8.7 Subroutine Name: PKQDM

1. Entry Point: PKQDM
2. Purpose: To calculate the material properties matrix, update the ECPTNL entry, and arrange the flow of element stiffness calculations for the QDMEM element.
3. Calling Sequence: CALL PKQDM

4.55.8.8 Subroutine Name: PKTRI1

1. Entry Point: PKTRI1
2. Purpose: To calculate the material properties matrix, update the ECPTNL entry, and arrange the flow of element stiffness calculations for the TRIA1 element.
3. Calling Sequence: CALL PKTRI1

4.55.8.9 Subroutine Name: PKTRI2

1. Entry Point: PKTRI2
2. Purpose: To calculate the material properties matrix, update the ECPTNL entry, and arrange the flow of element stiffness calculations for the TRIA2 element.
3. Calling Sequence: CALL PKTRI2

4.55.8.10 Subroutine Name: PKQAD1

1. Entry Point: PKQAD1
2. Purpose: To calculate the material properties matrix, update the ECPTNL entry, and arrange the flow of element stiffness calculations for the QUAD1 element.
3. Calling Sequence: CALL PKQAD1

4.55.8.11 Subroutine Name: PKQAD2

1. Entry Point: PKQAD2
2. Purpose: To calculate the material properties matrix, update the ECPTNL entry, and arrange the flow of element stiffness calculations for the QUAD2 element.
3. Calling Sequence: CALL PKQAD2

## MODULE FUNCTIONAL DESCRIPTIONS

### 4.55.8.12 Subroutine Name: PKTRM1

1. Entry Point: PKTRM1
2. Purpose: To generate element stress matrices for the TRMEM, TRIA1 and TRIA2 elements, and perform subcomputations for the PKQDM1 routine.
3. Calling Sequence: CALL PKTRM1 (NTYPE)

NTYPE  $\begin{cases} 0 = \text{TRMEM, TRIA1 or TRIA2} \\ 1 = \text{Subcomputations for the PKQDM1 routine} \end{cases}$

### 4.55.8.13 Subroutine Name: PKQDM1

1. Entry Point: PKQDM1
2. Purpose: To generate element stress matrices for the QDMEM, QUAD1 and QUAD2 elements.
3. Calling Sequence: CALL PKQDM1

### 4.55.8.14 Subroutine Name: PKTQ1

1. Entry Point: PKTQ1
2. Purpose: To generate element stress matrices for the TRIA1, TRIA2, QUAD1, and QUAD2 elements.
3. Calling Sequence: CALL PKTQ1 (NTYPE)

NTYPE  $\begin{cases} 1 = \text{TRIA1} \\ 2 = \text{TRIA2} \\ 3 = \text{QUAD1} \\ 4 = \text{QUAD2} \end{cases}$

### 4.55.8.15 Subroutine Name: PKTRQ2

1. Entry Point: PKTRQ2
2. Purpose: To perform final stress computations for the TRMEM and QDMEM elements.
3. Calling Sequence: CALL PKTRQ2 (NTYPE)

NTYPE  $\begin{cases} 1 = \text{TRMEM element} \\ 2 = \text{QDMEM element} \end{cases}$



4.55.8.16 Subroutine Name: PKTQ2

1. Entry Point: PKTQ2
2. Purpose: To perform final stress computations for the TRIA1, TRIA2, QUAD1, and QUAD2 elements.
3. Calling Sequence: CALL PKTQ2 (NPTS)

NPTS  $\begin{cases} 3 = \text{TRIA1 or TRIA2 elements} \\ 4 = \text{QUAD1 or QUAD2 elements} \end{cases}$

4.55.8.17 Subroutine Name: PKTRMS

1. Entry Point: PKTRMS
2. Purpose: To generate the element stiffness matrix for the TRMEM element and sub-computations for the PKQDMS routine.
3. Calling Sequence: CALL PKTRMS (NTYPE)

NTYPE  $\begin{cases} 0 = \text{TRMEM} \\ 1 = \text{Sub-computations for PKQDMS} \end{cases}$

4.55.8.18 Subroutine Name: PKQDMS

1. Entry Point: PKQDMS
2. Purpose: To generate the element stiffness matrix for the QDMEM element.
3. Calling Sequence: CALL PKQDMS

4.55.8.19 Subroutine Name: PKTRQD

1. Entry Point: PKTRQD
2. Purpose: To generate the element stiffness matrix for the TRIA1, TRIA2, QUAD1, or QUAD2 elements.
3. Calling Sequence: CALL PKTRQD (NTYPE)

## MODULE FUNCTIONAL DESCRIPTIONS

NTYPE  $\left\{ \begin{array}{l} 1 = \text{TRIA1} \\ 2 = \text{TRIA2} \\ 3 = \text{QUAD1} \\ 4 = \text{QUAD2} \end{array} \right.$

### 4.55.8.20 Subroutine Name: PKTRBS

1. Entry Point: PKTRBS
2. Purpose: To generate the element stiffness matrix subcalculations for the PKTRPL and PKQDPL routines.
3. Calling Sequence: CALL PKTRBS (IØPT)

IØPT  $\left\{ \begin{array}{l} 1 = \text{Subcomputations for PKQDPL} \\ 2 = \text{Subcomputations for PKTRPL} \end{array} \right.$

### 4.55.8.21 Subroutine Name: PKTRPL

1. Entry Point: PKTRPL
2. Purpose: To generate the element stiffness matrix for the TRIA1 and TRIA2 elements.
3. Calling Sequence: CALL PKTRPL

### 4.55.8.22 Subroutine Name: PKQDPL

1. Entry Point: PKQDPL
2. Purpose: To generate the element stiffness matrix for the QUAD1 and QUAD2 elements.
3. Calling Sequence: CALL PKQDPL

### 4.55.9 Design Requirements

The module was designed so that phase 1 and phase 2 can be executed in separate overlay segments.

Open core for phase 1 is defined at /PLA41X/ and for phase 2 at /PLA42X/. In phase 1 the single precision incremental displacement vector in unpacked form must be able to be contained in core. In phase 2, the open core requirements are the same as those for module SMA1 (see section 4.27.9.1) except that only four GINØ buffers are required during the principal loop of phase 2,



## FUNCTIONAL MODULE PLA4 (PIECEWISE LINEAR ANALYSIS - PHASE 4)

which processes the ECPTS and GPCT in a complementary manner. One GINØ buffer is defined for each of KGGNL, ECPTNL1, ECPTS and GPCT.

In addition to /PLA42E/, which is 100 words in length, subroutine PLA42 uses the following common blocks: a) /PLA42D/, which is 300 double precision words in length, and is used as a scratch storage for the module's element routines; b) /PLA425/, which is 325 single precision words in length, and is used as scratch storage for the module element routines; and c) /PLA42C/, which is a communication region for phase 2 of the task of the module. /PLA42C/ is defined as follows: CØMMØN/PLA42C/NPVT,GAMMA,GAMMAS,IPASS,ICSTM,NCSTM,IGPCT,NGPCT,IPØINT,NPØINT,I6X6K,N6X6K,CSTM,MPT,ECPTS,GPCT,DIT,KGGNL,ECPTØ,INRW,ØUTRW,EØR,NEØR,CLSRW,JMAX,FRØWIC,LRØWIC,NRØWSC,NLINKS,NWØRDS(40),IØVRLY(40),LINK(40),NØGØ

|  |   |  |
|--|---|--|
| GAMMA,GAMMAS   | - | The load increment ratios as defined in Equations 2 and 3 in section 4.54. |
| IPASS  | - | Number of the current pass through the PLA DMAP loop.                      |
| NPVT,ICSTM,NCSTM,IGPCT,<br>NGPCT,IPØINT,NPØINT,<br>I6X6K,N6X6K | } | - As defined in section 4.27.9.  |
| CSTM,MPT,ECTPS,GPCT,<br>DIT,KGGNL                              | } | - GINØ file numbers for their corresponding data blocks.                   |
| ECPTØ  | - | GINØ file number for the ECTPNL1 data block.                               |
| INRW,ØUTRW,...,<br>IØVRLY(40),LINK(40),NØGØ                    | } | - As defined in section 4.27.9.  |

The variables a) corresponding to GINØ file numbers, b) GINØ parameter options (e.g., INRW, ØUTRW), and c) NLINKS, IØVRLY, and NWØRDS, and NØGØ are set in the block data subprogram PLA4BD.

One scratch file is used, and all operations associated with stiffness matrix calculations are performed in double precision.

### 4.55.10 Diagnostic Messages

During phase 1, if the incremental displacement vector is null, user fatal error 2083 will occur.

During phase 2, error messages 3001, 3002, or 3003 may occur if the proper loading factors cannot be found on the PLFACT bulk data card image in the MPT. Other diagnostic messages for phase 2 are the same as those for module SMA1 (see section 4.27.10).

## FUNCTIONAL MODULE CASE (SIMPLIFY CASE CONTROL)

### 4.56 FUNCTIONAL MODULE CASE (SIMPLIFY CASE CONTROL)

#### 4.56.1 Entry Point: CASE

#### 4.56.2 Purpose

To remove looping considerations from later dynamics modules.

#### 4.56.3 DMAP Calling Sequence

CASE CASECC,PSDL/CASEXX/C,N,APPROACH/V,N,REPEAT/V,N,LOOP \$

#### 4.56.4 Input Data Blocks

CASECC - Case Control Data Table.

PSDL - Power Spectral Density List.

Note: PSDL is used only if APPROACH = FREQRESP and Random Analysis is selected in CASECC.

#### 4.56.5 Output Data Blocks

CASEXX - Case Control data table for dynamics problems.

Note: CASEXX cannot be purged.

#### 4.56.6 Parameters

APPROACH - Input-BCD-no default. Defines the approach to be used for looping criteria.

| <u>BCD Value</u> | <u>LOOP</u>                                 |
|------------------|---|
| STATICS          | NONE  |
| REIGEN           | NONE  |
| DS0              | NONE  |
| DS1              | NONE  |
| FREQRESP         | DIRECT INPUT MATRICES OR TRANSFER FUNCTIONS |
| TRANRESP         | LOADS                                       |
| BLK0             | NONE  |
| BLK1             | NONE  |
| CEIGEN           | DIRECT INPUT MATRICES OR TRANSFER FUNCTIONS |



## MODULE FUNCTIONAL DESCRIPTIONS

| <u>BCD Value</u> | <u>LOOP</u> |
|------------------|-------------|
| PLA              | NONE        |

REPEAT - Input and output-integer-set equal to zero outside of the DMAP loop by the PARAM module. -1 if no additional loops; + loop count if loops.

LOOP - Output-integer-default = -1. -1 if this is not a looping problem, 0 if this is a looping problem.

### 4.56.7 Method

The method of operation depends upon the input parameter APPRØACH.

#### 4.56.7.1 Transient Response

If APPRØACH = TRANRESP, CASECC is skipped over REPEAT records. If REPEAT = 0, REPEAT is set to 1. One record of CASECC is read and copied onto CASEXX. An attempt is made to read another record. If no more records exist, REPEAT is set to -1. Also, if this is the first entry to CASE (i.e., REPEAT = 1), LOOP is set to -1. If additional records exist, REPEAT and LOOP are set to 1.

#### 4.56.7.2 Complex Eigenvalue Analysis

If APPRØACH = CEIGEN, REPEAT records are skipped in CASECC. If REPEAT = 0, REPEAT is set to 1. One record of CASECC is read and copied onto CASEXX. The names of the Direct Input Matrices and Transfer Functions sets are saved. An attempt is made to read another record. If no more exist, REPEAT is set to -1. Also if this is the first entry (i.e., REPEAT = 1) LOOP is set to -1. If additional records exist, their Direct Input Matrices and Transfer Functions sets are compared to those saved. If they all agree, this record is copied onto CASEXX and the process is repeated. If they do not agree, REPEAT is incremented by 1, LOOP is set to 1, and CASE returns.

#### 4.56.7.3 Frequency Response

If APPRØACH = FREQRESP, the method used is the same as Complex Eigenvalue Analysis except a test is also made for frequency set selection changes. In addition, if RANDPS cards are selected, the selected set is read from PSDL and the unique subcase "id's" referenced are stored. Each subcase id copied onto CASEXX is compared to this list, and the entry is marked as found. If at the completion of CASE unmarked entries exist, the routine terminates with message 3033.

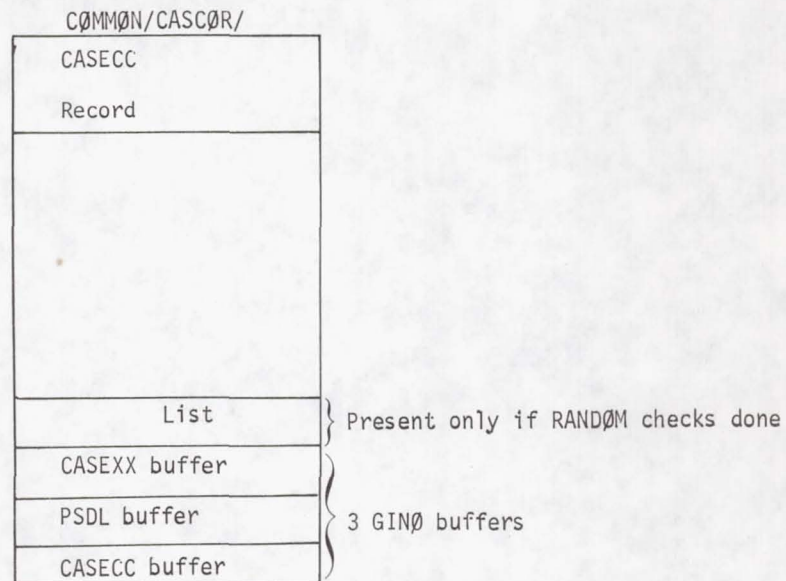
## FUNCTIONAL MODULE CASE (SIMPLIFY CASE CONTROL)

### 4.56.8. Subroutines

No auxiliary subroutines are used by CASE.

### 4.56.9 Design Requirements

Open core is defined at /CASCØR/.



### 4.56.10 Diagnostic Messages

If a case control record cannot be held in core, CASE will issue error message 3008.  
Message 3033 may be issued by CASE as outlined above.



FUNCTIONAL MODULE MTRXIN (MATRIX INPUT)

#### 4.57 FUNCTIONAL MODULE MTRXIN (MATRIX INPUT)

4.57.1 Entry Point: MTRXIN

#### 4.57.2 Purpose

MTRXIN has two purposes: (1) to provide a capability for direct input matrices as may occur in control systems in the dynamics Rigid Formats and, (2) to provide the DMAP user a capability of converting matrices input on DMIG bulk data cards to NASTRAN matrix format.

### 4.57.3 DMAP Calling Sequences

## 1. Dynamics Rigid Formats:

MTRXIN {CASEXX}, {BDP00L}, EQDYN,,TFP00L/{K2DPP}, {M2DPP}, {B2PP}/V,N,LUSETD/V,N,N0MAT1/  
CASECC}, {MATP00L}, {K2PP}, {M2PP}, {HB2PP}  
V,N,N0MAT2/V,N,NOMAT3 \$

## 2. DMAP Approach:

```

MTRXIN,      ,MATPOOL,EQEXIN,,/NAME1,NAME2,NAME3/V,N,LUSET/V,N,NOMAT1/V,N,NOMAT2/V,N,NOMAT3 $

```

#### 4.57.4 Input Data Blocks

CASECC - Case Control.

CASEXX - Case Control data table for dynamics problems.

MATPOOL - Data block containing matrices input on DMIG bulk data cards.

BDP00L - Hydroelastic boundary matrix tables.

EQDYN - Equivalence between external numbers and internal numbers, dynamics.

TFPOOL - Transfer Function Pool.

EQEXIN - Equivalence between external numbers and internal numbers.

Notes:

1. If CASECC is purged, the second purpose is assumed by MTRXIN.
2. EQDYN, EQEXIN, SIL and SILD may not be purged.

## MODULE FUNCTIONAL DESCRIPTIONS

### 4.57.5 Output Data Blocks

- K2DPP - Direct input differential stiffness matrix - p set.
- |       |   |   |                                   |
|-------|---|---|-----------------------------------|
| K2PP  | { | - | Direct input mass matrix - p set. |
| HK2PP |   |   |                                   |
- M2DPP - Direct input differential mass matrix - p set.
- M2PP - Direct input mass matrix - p set.
- |       |   |   |                                      |
|-------|---|---|--------------------------------------|
| B2PP  | { | - | Direct input damping matrix - p set. |
| HB2PP |   |   |                                      |
- |       |   |   |  |
|-------|---|---|--|
| NAME1 | { | - | The same names that appear on the DMIG cards, i.e., the DMIG matrix called NAME1 will be output on data block NAME1. |
| NAME2 |   |   |  |
| NAME3 |   |   |  |

Note: Any output data block may be purged.

### 4.57.6 Parameters

- LUSET - Input-integer-no default. Degrees of freedom in the g set. Used with EQEXIN and SIL.
- LUSETD - Input-integer-no default. Degrees of freedom in the p set. Used with EQDYN and SILD.
- NØMATi - Output-integer-no default. +1 if the i<sup>th</sup> output data block is generated, -1 otherwise.

### 4.57.7 DMAP Example

Assume the bulk data contain two DMIG matrices named M1 and M2 which reference grid and/or scalar points only. The following set of DMAP instructions will generate these two matrices in NASTRAN matrix format, multiply them together and print the result.

```

BEGIN
GP1      GEOM1,GEØM2/GPL,EQEXIN,GPDT,CSTM,BGPDT,SIL/V,N,LUSET/C,N,O/C,N,O  $
SAVE     LUSET $
MTRXIN,  ,MATPØØL,EQEXIN,  ,/M1,M2,/V,N,LUSET/V,N,NØM1/V,N,NØM2/C,N,O  $
SAVE     NØM1,NØM2 $
CØND     EXIT,NØM1 $
MPYAD    M1,M2,/PRØDUCT/C,N,O/C,N,1/C,N,1  $
MATPRN   PRODUCT,,,,// $
LABEL    EXIT $
END

```



4.57.8 Method

The first logical record in the Case Control data block is read into core, and the names of the requested DMIG matrices are fetched. If the Case Control data block is purged, FNAME is called to determine the names of the DMIG matrices from the names of the output data blocks. If the Case Control record was read, the transfer function set selection is fetched. If transfer matrices are requested, the TFP00L data block is opened, and the file is positioned to the requested set. Each transfer function matrix for which a corresponding direct input matrix exists is written on a scratch file. If no direct input matrix exists corresponding to a transfer function matrix, the transfer function matrix is written directly on the appropriate output data block. The transfer function matrices are written in NASTRAN matrix format by decoding the row and column numbers and calling BLDPK.

Upon completion of the writing of the transfer function matrices (if any), the second record of EQEXIN or EQDYN is read into core. The second word of each entry is converted into a scalar index number by dividing by 10. The MATP00L data block is opened. The following processing occurs:

1. The header information for the DMIG matrix is read. If an end-of-file is encountered, step (5) is executed. If the matrix is not requested, the remainder of the record is skipped and step (1) is repeated. Otherwise, step (2) is executed.
2. Each term in the matrix is read. The grid identification and component code are converted to a scalar index value by performing a binary search in EQEXIN or EQDYN in core. The scalar index forms a row position of the matrix. The row and column number (packed in one word) and the value for the term are stored in core. If core storage is exceeded, the terms are written on a scratch file.
3. When all terms have been read, converted and stored, the matrix is sorted by SORT. The matrix is now written in NASTRAN format by BLDPK.
4. If a transfer function is to be added to the DMIG matrix, the ADD routine is called to accomplish the matrix addition.
5. A test is made to determine if all requested matrices have been processed. If not, an error message is queued, and PEXIT is called. Otherwise, the module makes a normal exit.

## MODULE FUNCTIONAL DESCRIPTIONS

### 4.57.9 Design Requirements

#### 4.57.9.1 Allocation of Core Storage

Storage is required to hold the EQDYN or EQEXIN table (2 words per point in the problem) plus five GINØ buffers. Complete spill logic is provided for processing the DMIG matrices.

#### 4.57.9.2 Environment

The module MTRXIN consists of one subroutine, MTRXIN. Calls are made to the utility routine SØRT and matrix operation ADD. Open core is defined by /MTRXXX/. Seven scratch files are used.

#### 4.57.10 Diagnostic Messages

The following messages may be issued by MTRXIN:

2065, 2070, 2074.



## FUNCTIONAL MODULE GKAD (GENERAL K ASSEMBLER DIRECT)

### 4.58 FUNCTIONAL MODULE GKAD (GENERAL K ASSEMBLER DIRECT)

#### 4.58.1 Entry Point: GKAD

#### 4.58.2 Purpose

To assemble the dynamic stiffness, damping and mass matrices.

#### 4.58.3 DMAP Calling Sequence

GKAD    {HUSED}, GM, {HG0}, {HKA}, {HBAA}, {HRAA}, K4AA, {HK2PP}, M2PP, {HB2PP} /  
          {USED}, {G0}, {KA}, {BA}, {MA},  
          {HKDD}, {HBDD}, {HRDD}, GMD, {HGDD}, {HK2DD}, {HM2DD}, {HB2DD} /  
          {KDD}, {BDD}, {MDD}, {GDD}, {K2DD}, {M2DD}, {B2DD} /

#### 4.58.4 Input Data Blocks

HUSED  
USED    - Displacement set definitions table dynamics.

GM       - Multipoint constraint transformation matrix - m set.

G0       - Structural matrix partitioning transformation matrix.

KA       - Partition of stiffness matrix - a set.

BA       - Partition of damping matrix - a set.

MA       - Partition of mass matrix - a set.

K4AA     - Partition of structural damping matrix - a set.

K2PP     - Direct input stiffness matrix - p set.

M2PP     - Direct input mass matrix - p set.

B2PP     - Direct input damping matrix - p set.

HG0       - Heat matrix partitioning transformation matrix.

HKA       - Partition of the conductivity matrix - a set.

HBAA     - Partition of the capacity matrix - a set.

HRAA     - Partition of the radiation matrix - a set.

HK2PP    - Direct input conductivity matrix - p set.

HB2PP    - Direct input capacity matrix - p set.

## MODULE FUNCTIONAL DESCRIPTIONS

- Notes:
1. USETD cannot be purged.
  2. GM cannot be purged if  $MPCF1 \geq 0$ .
  3. GØ cannot be purged if  $ØMIT \geq 0$ .
  4. KAA cannot be purged if  $KDEKA \geq 0$ .
  5. BAA cannot be purged if  $NØBGG \geq 0$ .
  6. MAA may be purged.
  7. K4AA cannot be purged if  $NØK4GG \geq 0$ .
  8. K2PP cannot be purged if  $NØK2PP \geq 0$ .
  9. M2PP cannot be purged if  $NØM2PP \geq 0$ .
  10. B2PP cannot be purged if  $NØB2PP \geq 0$ .

### 4.58.5 Output Data Blocks

- KDD - Dynamic stiffness matrix - d set.
- BDD - Dynamic damping matrix - d set.
- MDD - Dynamic mass matrix - d set.
- GMD - Multipoint constraint transformation matrix - dynamics.
- HGØD  
GØD - Omitted coordinate transformation matrix - dynamics.
- K2DD - Direct input stiffness matrix - d set.
- M2DD - Direct input mass matrix - d set.
- B2DD - Direct input damping matrix - d set.
- HKDD - Dynamic conductivity matrix - d set.
- HBDD - Dynamic capacity matrix - d set.
- HRDD - Dynamic radiation matrix - d set.
- HK2DD - Direct input conductivity matrix - d set.
- HM2DD - Direct input radiation matrix - d set.
- HB2DD - Direct input capacity matrix - d set.

- Notes:
1. GMD cannot be purged if  $MPCF \geq 0$ .
  2. GØD cannot be purged if  $ØMIT \geq 0$ .
  3. K2DD cannot be purged if  $NØK2PP \geq 0$ .
  4. M2DD cannot be purged if  $NØM2PP \geq 0$ .
  5. B2DD cannot be purged if  $NØB2PP \geq 0$ .



## FUNCTIONAL MODULE GKAD (GENERAL K ASSEMBLER DIRECT)

### 4.58.6 Parameters

- TYPE    - Input-BCD-no default. If TYPE = TRANSIENT the transient equations are used; otherwise the frequency response equations are used.
- APP     - Input-BCD-no default. If APP = FØRCE the p set = d set; otherwise p's are reduced to d's by removing m's, s's, and o's.
- FØRM    - Input-BCD-no default. If FØRM = MØDAL, KDD and BDD are not computed. MDD is not computed unless MØDACC  $\geq$  0.
- G       - Input-real-default = 0.0. G is the coefficient of K4DD if TYPE  $\neq$  TRANSIENT. G/W3 is coefficient of K1DD if TYPE = TRANSIENT.
- W3      - Input-real-default = 0.0. If TYPE = TRANSIENT G/W3 is the coefficient of K1DD. If W3 = 0.0 K1DD is not used.
- W4      - Input-real-default = 0.0. 1.0/W4 is the coefficient of K4DD if TYPE = TRANSIENT. If W4 = 0.0 K4DD is not used.
- NØK2PP - Input-integer-no default. NØK2PP  $\geq$  0 indicates presence of K2PP.
- NØM2PP - Input-integer-no default. NØM2PP  $\geq$  0 indicates presence of M2PP.
- NØB2PP - Input-integer-no default. NØB2PP  $\geq$  0 indicates presence of B2PP.
- MPCF1   - Input-integer-no default. MPCF1  $\geq$  0 indicates presence of GM.
- SINGLE   - Input-integer-no default. SINGLE  $\geq$  0 indicates presence of single-point constraints.
- ØMIT    - Input-integer-no default. ØMIT  $\geq$  0 indicates presence of GØ.
- NØUE    - Input-integer-no default. NØUE  $\geq$  0 indicates presence of extra points.
- NØK4GG - Input-integer-no default. NØK4GG  $\geq$  0 indicates presence of K4AA.
- NØBGG   - Input-integer-no default. NØBGG  $\geq$  0 indicates presence of BAA.
- KDEKA   - Input-integer-no default. KDEKA  $\geq$  0 indicates presence of MAA and KAA.
- MØDACC - Input-integer-default = -1. MØDACC  $\geq$  0 requests computation of MDD (meaningful only if FØRM = MØDAL)

# MODULE FUNCTIONAL DESCRIPTIONS

## 4.58.7 Method

If extra points are present ( $N\emptyset UE \geq 0$ ) and multipoint constraints or omitted coordinates are present ( $MPCF1 \geq 0$  or  $\emptyset MIT \geq 0$ ), then

$$GM \Rightarrow GMD, \quad (1)$$

and

$$G\emptyset \Rightarrow G\emptyset D. \quad (2)$$

Subroutine GKAD1A performs these tasks.

If direct input matrices are present and m's, s's or o's are present, the direct input matrices are reduced from the p set to the d set. Let  $[D_{pp}^2]$  be a direct input matrix,

$$[D_{pp}^2] = [K_{pp}^2], [M_{pp}^2] \text{ or } [B_{pp}^2]$$

1. If m's are present,

$$[D_{pp}^2] \Rightarrow \begin{bmatrix} \bar{D}_{nn}^2 & \vdots & \bar{D}_{nm}^2 \\ \vdots & \ddots & \vdots \\ \bar{D}_{mn}^2 & \vdots & \bar{D}_{mm}^2 \end{bmatrix}. \quad (3)$$

(The e coordinates are included with the n coordinates). Then compute:

$$[D_{nn}^2] = [\bar{D}_{nn}^2] + [\bar{D}_{nm}^2] [G_m^d] + [G_m^d]^T [\bar{D}_{mn}^2] + [G_m^d]^T [\bar{D}_{mm}^2] [G_m^d].$$

2. If s's are present,

$$[D_{ff}^2] \Rightarrow \begin{bmatrix} D_{ff}^2 & \vdots & D_{fs}^2 \\ \vdots & \ddots & \vdots \\ D_{sf}^2 & \vdots & D_{ss}^2 \end{bmatrix}, \quad (5)$$

where only  $[D_{ff}^2]$  is saved. The e coordinates are included with the f coordinates.

FUNCTIONAL MODULE GKAD (GENERAL K ASSEMBLER DIRECT)

3. If o's are present, first partition  $[D_{ff}^2]$

$$[D_{ff}^2] \Rightarrow \begin{bmatrix} \bar{D}_{dd}^2 & \vdots & \bar{D}_{do}^2 \\ \hline \bar{D}_{od}^2 & \vdots & \bar{D}_{oo}^2 \end{bmatrix}, \quad (6)$$

then:

$$[D_{dd}^2] = [\bar{D}_{dd}^2] + [\bar{D}_{do}^2] [G_o^d] + [G_o^d]^T [D_{od}] + [G_o^d]^T [\bar{D}_{oo}^2] [G_o^d]. \quad (7)$$

Steps 1 through 3 are done for K2PP, M2PP and B2PP, using subroutines GKAD1C and GKAD1D.

If FØRM = MØDAL and MØDACC < 0, GKAD is done. If not, the a set matrices are expanded to the d set by adding zeros at extra points. Let  $[D_{aa}]$  be an a set matrix. Then,

$$\begin{bmatrix} D_{aa} & \vdots & 0 \\ \hline 0 & \vdots & 0 \end{bmatrix} \Rightarrow [D_{dd}^1]. \quad (8)$$

The above step is done for KAA, BAA, MAA, and K4AA and is performed in subroutine GKAD1B.

Compute KDD, BDD and MDD.

1. For Frequency Response or Complex Eigenvalue Analysis (TYPE ≠ TRAN),

$$[K_{dd}] = (1+iG) [K_{dd}^1] + [K_{dd}^2] + i [K_{dd}^{14}], \quad (9)$$

$$[B_{dd}] = [B_{dd}^1] + [B_{dd}^2], \quad (10)$$

$$[M_{dd}] = [M_{dd}^1] + [M_{dd}^2]. \quad (11)$$



# MODULE FUNCTIONAL DESCRIPTIONS

2. For Transient Analysis (TYPE = TRAN),

$$[K_{dd}] = [K_{dd}^1] + [K_{dd}^2] , \quad (12)$$

$$[B_{dd}] = [B_{dd}^1] + [B_{dd}^2] + \frac{G}{W3} [K_{dd}^1] + \frac{1.0}{W4} [K_{dd}^{14}] , \quad (13)$$

$$[M_{dd}] = [M_{dd}^1] + [M_{dd}^2] . \quad (14)$$

If W3 or W4 is zero, the corresponding matrices are ignored.



#### 4.58.8 Subroutines

GKAD uses matrix utility routines SSG2B, SSG2C, CALCV, MERGE, UPART, and MPART. Descriptions for these routines can be found in Section 3.

##### 4.58.8.1 Subroutine Name: GKAD1A

1. Entry Point: GKAD1A
2. Purpose: To expand GM or GØ to d size matrices:

$$[G_m \mid 0] \Rightarrow [G_m^d] \quad (15)$$

3. Calling Sequence: CALL GKAD1A (USETD,GØ,GØD,SCR1,UE,UA,UNE)
  - USETD - GINØ file number of USETD - integer - input.
  - GØ - GINØ file number of GØ - integer - input.
  - GØD - GINØ file number of GØD - integer - input.
  - SCR1 - GINØ file number of scratch file - integer - input.
  - UE - Pointer to UE bit in USETD word - integer - input.
  - UA - Pointer to UA bit in USETD word - integer - input.
  - UNE - Pointer to UNE bit in USETD word - integer - input.

##### 4.58.8.2 Subroutine Name: GKAD1B

1. Entry Point: GKAD1B
2. Purpose: To expand a set matrices to d set size.
3. Calling Sequence: CALL GKAD1B (USETD,KAA,MAA,BAA,K4AA,K1DD,M1DD,B1DD,K41DD,UA,UE,UD,SCR1)
  - USETD - GINØ file number of USETD - integer - input.
  - KAA - GINØ file number of KAA - integer - input.
  - MAA - GINØ file number of MAA - integer - input.
  - BAA - GINØ file number of BAA - integer - input.
  - K4AA - GINØ file number of K4AA - integer - input.

## MODULE FUNCTIONAL DESCRIPTIONS

K1DD - GINØ file number of K1DD - integer - input.  
M1DD - GINØ file number of M1DD - integer - input.  
B1DD - GINØ file number of B1DD - integer - input.  
K41DD - GINØ file number of K41DD - integer - input.  
SCR1 - GINØ file number of scratch file - integer - input.  
UA - Pointer to UA bit in USETD word - integer - input.  
UE - Pointer to UE bit in USETD word - integer - input.  
UD - Pointer to UD bit in USETD word - integer - input.

### 4.58.8.3 Subroutine Name: GKAD1C

1. Entry Point: GKAD1C
2. Purpose: To initialize GKAD1D.
3. Calling Sequence: CALL GKAD1C (GMD,GØD,SCR1,SCR2,SCR3,SCR4,SCR5,SCR6,USETD)  
GMD,GØD,USETD are GINØ file numbers of their respective data blocks - integer - input.  
SCR1,...,SCR6 are GINØ file numbers of six scratch files - integer - input.

### 4.58.8.4 Subroutine Name: GKAD1D

1. Entry Point: GKAD1D
2. Purpose: To reduce "2PP" matrices to "2DD" matrices.
3. Calling Sequence: CALL GKAD1D (K2PP,K2DD)  
K2PP - GINØ file number of input matrix - integer - input.  
K2DD - GINØ file number of reduced matrix - integer - input.

### 4.58.9 Design Requirements

Six scratch files are necessary. Open core for GKAD1A and GKAD1B is defined at /GKADA1/.  
Open core for GKAD1C and GKAD1D is defined at /GKADC1/.

### 4.58.10 Diagnostic Messages

None



## FUNCTIONAL MODULE CEAD (COMPLEX EIGENVALUE ANALYSIS - DISPLACEMENT)

### 4.59 FUNCTIONAL MODULE CEAD (COMPLEX EIGENVALUE ANALYSIS - DISPLACEMENT)

#### 4.59.1 Entry Point: CEAD

#### 4.59.2 Purpose

To solve the equation

$$([M]p^2 + [B]p + [K])\{u\} = \{0\} \quad (1)$$

for the eigenvalues  $p$  and the associated eigenvectors  $\{u\}$  where  $[M]$ ,  $[B]$  and  $[K]$  are mass, damping and stiffness matrices respectively.

#### 4.59.3 DMAP Calling Sequence

CEAD KDD,BDD,MDD,EED,CASECC/PHID,CLAMA,ØCEIGS,PHIDL/V,N,NFØUND \$

#### 4.59.4 Input Data Blocks

- KDD - Dynamic stiffness matrix - d set.
- BDD - Dynamic damping matrix - d set.
- MDD - Dynamic mass matrix - d set.
- EED - Eigenvalue Extraction Data.
- CASECC - Case Control Data Table.

#### Notes:

1. EED must be present.
2. CASECC must be absent when a substructure modal reduce is being performed. In all other cases it must be present.
3. At least one of KDD, BDD and MDD must be present.

#### 4.59.5 Output Data Blocks

- PHID - Complex eigenvectors in the d set.
- CLAMA - Complex eigenvalue table.
- ØCEIGS - Complex eigenvalue summary table.
- PHIDL - Left complex eigenvectors in the d set.



## MODULE FUNCTIONAL DESCRIPTIONS

### Notes:

1. PHID, CLAMA and ØCEIGS must be present.
2. PHIDL may be purged if the left-hand vectors are not desired.

### 4.59.6 Parameters

NFØUND - Output-integer-no default. NFØUND indicates the number of eigenvalues found.

If none were found, NFØUND is set to -1.

### 4.59.7 Method

The Complex Eigenvalue Analysis Module calculates the eigenvalues and eigenvectors for a general system which may have complex terms in the mass, damping, and stiffness matrices. The eigenvectors are scaled according to the user-requested normalization scheme. Modal masses are not calculated since they will, in general, be complex, and their value is rather dubious. The form of the problem solved by the Complex Eigenvalue Analysis Module is given in Equation 1.

The eigenvalues  $p$  and the eigenvectors  $\{u\}$  are always treated as complex. These data are related to the  $u_d$  displacements if a direct formulation is used or are related to the  $u_h$  displacements if a modal formulation is used. The method to be used and the necessary data are selected by calling for one ID number in the EED data block. A set of EED data which defined either the Determinant Method, the Inverse Power Method, the Tridiagonal Reduction Method or the Hessenburg Method must be used. Subroutine CDETM, CFEER, CINVPR, or HESS1 is called to solve the eigenvalue problem (see subroutine descriptions below for method details. The eigenvalues and associated vectors are sorted by the magnitude of the imaginary part of the eigenvalue with all positives listed ahead of all negatives. (Subroutine CEAD1A).

### 4.59.8 Subroutines

The subroutines used by CEAD can be divided into six groups: 1) those used by CEAD; 2) those used for the Inverse Power Method; 3) those used by the Determinant Method; 4) those used by the Hessenburg Method; 5) those used by the Tridiagonal Reduction (FEER) Method; and 6) general utility routines. The descriptions of the utility routines can be found in Section 3.

# FUNCTIONAL MODULE CEAD (COMPLEX EIGENVALUE ANALYSIS - DISPLACEMENT)

| <u>CEAD</u> | <u>Determinant</u> | <u>Hessenberg</u> | <u>Inverse Power</u> |        | <u>FEER</u> |        | <u>General</u> |
|-------------|--------------------|-------------------|----------------------|--------|-------------|--------|----------------|
| CEAD1A      | CDETM              | ALLMAT            | CDIFBS               | CMTIMU | CFCNTL      | CFE1MY | ADD            |
| CLVEC       | CDETM2             | CFACTR            | CDIVID               | CNØRM  | CFEER1      | CFE2MY | CDCØMP         |
|             | CDETM3             | CFBSØR            | CINFBS               | CNØRM1 | CFEER2      | CF1FBS | PRELØC         |
|             | CDTFBS             | HESS1             | CINVPR               | CSQRTX | CFEER3      | CF2FBS |                |
|             | CSQRTN             | HESS2             | CINVP1               | CSUB   | CFEER4      | CFNØR1 |                |
|             | CSUMM              | MERGED            | CINVP2               | CXTRNY | CFER3D      | CFNØR2 |                |
|             |                    |                   | CINVP3               | ØRTHØ  | CFER3S      | CF1ØRT |                |
|             |                    |                   |                      |        | CFE1AØ      | CF2ØRT |                |
|             |                    |                   |                      |        | CFE2AØ      |        |                |

## 4.59.8.1 Subroutine Name: CEAD1A

1. Entry Point: CEAD1A
  2. Purpose: To sort the eigenvalues and the right and left eigenvectors.
  3. Calling Sequence: CALL CEAD1A (LAMAI,PHII,PHIIL,LAMAØ,PHIØ,PHIØL,NFØUND,NVECT,CAPP)
- LAMAI - GINØ file number of unsorted eigenvalues - integer - input.
- PHII - GINØ file number of unsorted eigenvectors - integer - input.
- PHIIL - GINØ file number of unsorted left eigenvectors (Inverse Power Method only) - integer - input.
- LAMAØ - GINØ file number of data block CLAMA - integer - input.
- PHIØ - GINØ file number of data block PHID - integer - output.
- PHIØL - GINØ file number of data block PHIDL - integer - output.
- NFØUND - Number of eigenvalues found - integer - input.
- NVECT - Number of eigenvectors found - integer - input.
- CAPP - Method - BCD - input.

## 4.59.8.2 Subroutine Name: CINVPR

1. Entry Point: CINVPR
  2. Purpose: CINVPR is the main driver for the Complex Inverse Power Method of eigenvalue extraction.
  3. Calling Sequence: CALL CINVPR (EED,METHØD,NFØUND)
- CØMMØN / CINVPX / K(7),M(7),B(7),LAM(7),PHI(7),EIGSUM,SCRFIL(11),NØREG,EPS,REG(7,10),PHIDLI
- CØMMØN / CINVX / Z(1)



# MODULE FUNCTIONAL DESCRIPTIONS

- K,M,B - Input matrix control blocks for the stiffness, mass, and damping matrices (K), [M], and [B].
- LAM,PHI - Matrix control blocks for the output eigenvalue and eigenvector files.
- EIGSUM - The output eigenvalue summary file.
- SCRFIL(11) - Eleven scratch files available to Inverse Power
- NØREG - Number of regions input to CINVPR.
- EPS - Convergence criterion.
- REG(7,10) - Storage space for up to 10 region parameters.
- PHIDLI - GINØ file number of unsorted left eigenvector file.
- Z(1) - Open core for CINVPR.

# FUNCTIONAL MODULE CEAD (COMPLEX EIGENVALUE ANALYSIS - DISPLACEMENT)

┌

3. Calling Sequence: CALL CINV2

COMMON /CINVPX/ DUM(36),A,XX,L,U,SCR1,SCR2,SCR3,LL,UU

COMMON /CINVXX/DUMM(4),SWITCH

COMMON /CINV2X/Z(1)

A - GINØ file number for the input matrix.

L,U - GINØ file number for the lower and upper triangular factors output from CDCØMP.

SCR1,SCR2,SCR3 - Three scratch files used by CDCØMP.

LL,UU - GINØ file numbers for alternate storage of L and U.

SWITCH -  $\left\{ \begin{array}{l} 0, \text{ store factors on L and U.} \\ 1, \text{ store factors on LL and UU.} \\ -204, \text{ store factors on LL and UU, and recompute length of open core.} \\ \text{(Left-hand eigenvector calculations only)} \end{array} \right.$

Z(1) - Area of open core used by CDCØMP.



# MODULE FUNCTIONAL DESCRIPTIONS

## 4.59.8.5 Subroutine Name: CINVP3

1. Entry Point: CINVP3

2. Purpose: To solve for a complex eigenvalue and eigenvector using the Inverse Power Method.

3. Calling Sequence: CALL CINVP3

COMMON /CINVPX/K(7),M(7),B(7),LAM(7),PHI(7),XXX,SCRFIL(11)

COMMON /CINV3X/Z(1)

See section 4.59.8.2 above for details on /CINVPX/.

Z(1) - Area of open core available in CINVP3.

4. Method: The logic flow and the mathematical equations are essentially identical to INVP3, with the following exceptions. The eigenvalues and eigenvectors are found corresponding to the matrix equation

$$(\lambda^2[M] + \lambda[B] + [K]) [\Phi] = [0] \quad (3)$$

where the iteration equation is given by

$$(\lambda_o^2[M] + \lambda_o[B] + [K]) \{W_n\} = -([B] + \lambda_o[M]) \{u_{n-1}\} - [M] \{v_{n-1}\} \quad (4)$$

with

$$\{\bar{u}_n\} = \frac{1}{c_n} \{W_n\}, \quad (5)$$

$$\{\bar{v}_n\} = \lambda_o \{\bar{u}_n\} + \frac{1}{c_n} \{u_{n-1}\}, \quad (6)$$

$$\{u_n\} = \{\bar{u}_n\} - \sum_i \alpha_i \{\Phi_i\}, \quad (7)$$

$$\{v_n\} = \{\bar{v}_n\} - \sum_i \alpha_i \lambda_i \{\Phi_i\}, \quad (8)$$

and

$$\alpha_i = \frac{\{\Phi_i\}^T [\lambda_i[M] \{\bar{u}_n\} + [M] \{\bar{v}_n\} + [B] \{\bar{u}_n\}]}{\{\Phi_i\}^T (2\lambda_i [M] + [B]) \{\Phi_i\}}. \quad (9)$$

Γ

where

$\Phi_i$  = Previously extracted right-hand vector,

$\bar{\Phi}_i$  = Previously extracted left-hand vector,

$C_n$  = Largest element (in magnitude) of  $\{W_n\}$ , and

$\lambda_i$  = Previously extracted eigenvalue.

The above equations replace Equations 19 through 22 in section 4.48. The calculation of the remaining equations remains the same except for the use of complex arithmetic. The left eigenvector is obtained by decomposing Equation 3 with  $\lambda_0 - \lambda_i$  and using CDIFBS to make the appropriate substitution using the factors from CDCOMP.

5. Design Requirements: CINVPS requires fourteen complex double precision vectors in core plus four GINØ buffers.

#### 4.59.8.6 Subroutine Name: CNØRM

1. Entry Point: CNØRM

2. Purpose: To normalize successive iterated vectors such that the maximum element is equal to unity, and to return the normalizing divisor.

3. Calling Sequence: CALL CNØRM (X,DIV)

X - Input vector to be normalized.

DIV - Divisor which was used to normalize the vector corresponding to the argument X.

#### 4.59.8.7 Subroutine Name: CNØRM1

1. Entry Point: CNØRM1

2. Purpose: To normalize a complex vector such that the largest magnitude of an element is equal to one.

3. Calling Sequence: CALL CNØRM (X,N)

X - Vector to be normalized.

N - Length of the vector (complex terms).



## MODULE FUNCTIONAL DESCRIPTIONS

### 4.59.8.8 Subroutine Name: CINFBS

1. Entry Point: CINFBS
2. Purpose: To perform the forward-backward substitution necessary to solve an iteration of the Inverse Power Method.

3. Calling Sequence: CALL CINFBS (X,Y,BUF)

COMMON /CINFBX/L(7),U(7)

L,U - Matrix control blocks for the factors output from CDCOMP.

X - Complex double precision input vector.

Y - Complex double precision solution vector.

BUF - GINØ buffer.

4. Method: CINFBS is a stripped down version of GFBS. Both vectors reside in core, and only complex double precision arithmetic is used.

### 4.59.8.9 Subroutine Name: CDIFBS

1. Entry Point: CDIFBS
2. Purpose: To perform the forward-backward substitution necessary to solve for the left eigenvector.

3. Calling Sequence: CALL CDIFBS (X,BUF)

COMMON /CINVPX/DUM(41),UPRTRI,XXX,LØWTTRI

UPRTRI,LØWTTRI - Files containing the upper and lower triangular factors output from CDCOMP.

X - The output complex double precision left eigenvector.

BUF- GINØ buffer used by CDIFBS.

4. Method: CDIFBS actually solves the system of equations

$$[A]^T \{x\} = \{y\}, \quad (10)$$

where [A] has been decomposed into  $[A] = [L][U]$ . To solve the transpose problem we have that

$$[A]^T = ([L] [U])^T = [U]^T [L]^T \quad (11)$$

so that

$$[U]^T [L]^T \{x\} = \{y\}. \quad (12)$$

CDIFBS is a modified form of GFBS which does the forward pass on [U] and the backward pass on [L]. All arithmetic operations are complex double precision.

#### 4.59.8.10 Subroutine Name: CMTIMU

1. Entry Point: CMTIMU
2. Purpose: To pre-multiply a vector {y} by a matrix to obtain a vector {x}.
3. Calling Sequence: CALL CMTIMU (Y,X,FILE,BUF)

COMMON /CINVPX/DUM(7),M(7)

FILE - If FILE = 0, form {x} = [M]{y}.

FILE ≠ 0, form {x} = [A] {y}, where [A] is the matrix on FILE.

X,Y - Complex double precision vectors.

BUF - GINØ buffer.

#### 4.59.8.11 Subroutine Name: CXTRNY

1. Entry Point: CXTRNY
2. Purpose: To form the inner product of two complex vectors, {x} and {y}

$$a = \{x\}^T \{\bar{y}\}, \quad (13)$$

where  $\{\bar{y}\}$  denotes a vector all of whose components are the complex conjugates of {y}.

3. Calling Sequence: CALL CXTRNY (X,Y,A)

COMMON /CINVPX/XX,N

N - Length of the vectors.

X,Y - Complex double precision vectors.

A - Complex double precision value of the inner product of {x} and {y}.



# MODULE FUNCTIONAL DESCRIPTIONS

## 4.59.8.12 Subroutine Name: CSUB

1. Entry Point: CSUB
2. Purpose: To evaluate the vector equation

$$\{z\} = a\{x\} - b\{y\}, \quad (14)$$

where  $\{x\}$ ,  $\{y\}$ ,  $a$  and  $b$  may be complex.

3. Calling Sequence: CALL CSUB (X,Y,Z,A,B)

COMMON /CINVPX/XXX,N

- N - Length of the vectors  $\{x\}$  and  $\{y\}$ .
- X,Y,Z - Complex double precision vectors.
- A,B - Complex double precision scalar multipliers.

## 4.59.8.13 Subroutine Name: ORTHO

1. Entry Point: ORTHO
2. Purpose: To orthogonalize a vector with respect to all previously extracted vectors.
3. Calling Sequence: CALL ORTHO (U,V,X1,X2,X3,X4,X5,NZ,BUF1,BUF2,BUF3,BUF4)

COMMON /CINVPX/K(7),M(7),B(7),LAMBDA(7),PHI(7),XXX,SCRFIL(10)

COMMON /CINVXX/DUM(19),NRORTS

See section 4.59.8.2 for /CINVPX/ details.

- NRORTS - Number of eigenvectors already extracted.
- U,V - Input-current vectors - Output - orthogonalized vectors.
- X1,...,X5 - Storage space for five complex double precision vectors.
- NZ - The number of words of core available to ORTHO.
- BUF1,  $\left. \begin{matrix} \dots \\ \text{BUF4} \end{matrix} \right\}$  - Four GIN buffers.

4. Method: ORTHO solves the equations

$$\{u_n\} = \{u_n\} - \sum_i \alpha_i \lambda_i \{\Phi_i\}, \quad (15)$$

$$\{v_n\} = \{v_n\} - \sum_i \alpha_i \lambda_i \{\phi_i\}, \quad (16)$$

where

$$\alpha_i = \frac{\{\bar{\phi}_i\}^T [\lambda_i [M] \{u_n\} + [M] \{v_n\} + [B] \{u_n\}]}{\{\bar{\phi}_i\}^T [2 \lambda_i [M] + [B]] \{\phi_i\}} \quad (17)$$

and

$\{\bar{\phi}_i\}$  = Previously found left eigenvectors.

$\{\phi_i\}$  = Previously found right eigenvectors.

$\lambda_i$  = Previously found eigenvalues.

Note that the demoninator of equation 17 is constant with respect to the current iterate  $u_n$  and  $v_n$ . Thus it is computed once for each vector and saved on the left vector scratch file in place of the left vector.



## MODULE FUNCTIONAL DESCRIPTIONS

### 4.59.8.14 Subroutine Name: CDETM

1. Entry Point: CDETM
2. Purpose: To solve the complex eigenvalue problem by the Determinant Method.
3. Calling Sequence: CALL CDETM (METHØD,EED,M,B,K,LAMA,PHID,ØCEIGS,NFØUND,SCR1,SCR2,SCR3,SCR4,SCR5,SCR6,SCR7,SCR8)

METHØD            - ID of an EIGC card for the Determinant Method - integer - input.

EED,ØCEIGS, } - GINØ file numbers of their respective data blocks - integer - input.  
M,B,K

LAMA            - GINØ file number of temporary eigenvalue storage file - integer - input.

PHID            - GINØ file number of temporary eigenvector storage file - integer - input.

NFØUND           - Number of eigenvalues found - integer - output.

SCR1,SCR2, } - GINØ file numbers of 8 scratch files - integer - input.  
...,SCR8

4. Method: The overall flow and theoretical considerations of the Determinant Method are explained in section 4.88. Two refinements are made in CDETM. The first is the handling of multiple search regions, which allows the user to control the distribution of starting points in the complex plane. See the EIGC bulk data card description in section 2 of the User's Manual for further details. The second is the use of the EIGP card to define poles which will be swept from the determinant as if they were previously accepted eigenvalues. This allows the user to prevent convergence to known or already extracted eigenvalues.

5. Design Requirements: CDETM requires two complex double precision d set vectors plus one GINØ buffer in core.

### 4.59.8.15 Subroutine Name: CDETM2

1. Entry Point: CDETM2
  2. Purpose: To arrange 3 starting points in order of the magnitude of the determinant.
  3. Calling Sequence: CALL CDETM2(P,D,IP,PR,PI,DR,DI,IPS)
- P    - Three starting point values - input-complex double precision.

FUNCTIONAL MODULE CEAD (COMPLEX EIGENVALUE ANALYSIS - DISPLACEMENT)

- D - Scaled determinants at P - input-complex double precision.
- IP - Scale factors for D - input - integer.
- PR - Real parts of the reordered starting points - output-double precision.
- PI - Imaginary parts of the reordered starting points - output-double precision.
- DR - Real parts of the reordered determinants - output-double precision.
- DI - Imaginary parts of the reordered determinants - output-double precision.
- IPS - Scale factors of the reordered determinants - output - integer.

4.59.8.16 Subroutine Name: CSUMM

1. Entry Point: CSUMM
2. Purpose: To add two scaled complex numbers together.
3. Calling Sequence: CALL CSUMM (D1,D2,ID1,D3,D4,ID2,D5,D6,ID3)

The arguments are defined in the following equation:

$$(D1,D2) \times 10^{ID1} + (D3,D4) \times 10^{ID2} = (D5,D6) \times 10^{ID3}, \quad (18)$$

where all Di's are double precision.

4.59.8.17 Subroutine Name: CDTFBS

1. Entry Point: CDTFBS
2. Purpose: To solve for the eigenvector given the decomposed impedance matrix.
3. Calling Sequence: CALL CDTFBS (F,EV,BUFFER(1),FU,NRØW)

- F - Applied complex load vector - input-complex double precision.
- EV - Eigenvector - output- complex double precision.
- BUFFER(1) - GINØ buffer.
- FU - Matrix control block for [U] - integer - input.
- NRØW - Order of problem - integer - input.



# MODULE FUNCTIONAL DESCRIPTIONS

## 4.59.8.18 Subroutine Name: CDETM3

1. Entry Point: CDETM3
2. Purpose: To rescale a scaled complex number.
3. Calling Sequence: CALL CDETM3(D1,D2,ID1)

Let  $\overline{D1}$ ,  $\overline{D2}$ ,  $\overline{ID1}$  be the input values of D1, D2, ID1. On return from CDETM3

$$(D1,D2) \times 10^{ID1} = (\overline{D1},\overline{D2}) \times 10^{\overline{ID1}}, \quad (20)$$

and

$$1.0 \leq |(D1,D2)| \leq 10.0, \quad (21)$$

where all Di's are double precision.

## 4.59.8.19 Subroutine Name: CDIVID

1. Entry Point: CDIVID
2. Purpose: To divide a complex vector by a complex number.
3. Calling Sequence: CALL CDIVID (DIV,V,V1,NV)

where V is a complex D.P. vector of length NV to be divided by DIV and the answer put in V1.

## 4.59.8.20 Subroutine Name: HESS1

1. Entry Point: HESS1
2. Purpose: HESS1 is the overall driver for the upper Hessenburg method of complex eigenvalue extraction. CEAD will call HESS1 if sufficient core exists ( $6N^2 + 8N$ ) to use this method where N is the order of the reduced problem. Otherwise it will select complex inverse power. If the B matrix is not null  $N = 2*N$ .
3. Calling Sequence: CALL HESS1 (KDD,MDD,CLAMA,PHID,ØCEIGS,NFØUND,NVECD,BDD,SCR1,SCR2,SCR3,SCR4,SCR5,SCR6,SCR7,EED,METHØD)

# FUNCTIONAL MODULE CEAD (COMPLEX EIGENVALUE ANALYSIS - DISPLACEMENT)

where

KDD, MDD, CLAMA, PHID, ØCEIGS, BDD, SCR1-7, and EED are the GINØ file numbers of their respective data blocks.

NFØUND - integer-output-NFØUND is the number of eigenvalues found.

NVECD - integer-output-NVECD is the number of eigenvalues output.

METHØD - integer-input-METHØD is the EIGC Id selected in CASECC by the CMETHØD card.

4. Method: HESS1 transforms the eigenvalue problem  $[P^2[MDD] + P[BDD] + [KDD]] \{\Phi_D\} = \{0\}$  into  $[k^2 I + A]\{\Phi_{D1}\} = \{0\}$  according to the following procedures:

If BDD = 0

$$[A] = [MDD]^{-1} [KDD]$$

$$P = \sqrt{k^2} \text{ (with positive Im)}$$

$$\{\Phi_D\} = \{\Phi_{D1}\}$$

If BDD  $\neq$  0

$$A = \left[ \begin{array}{c|c} 0 & -I \\ \hline MDD^{-1} KDD & MDD^{-1} BDD \end{array} \right]$$

$$P = k^2$$

$$\{\Phi_{D1}\} = \left\{ \begin{array}{c} \Phi_D \\ 0 \end{array} \right\}$$

MDD must be non-singular and a special case exists if MDD is the identity (Form = 8).

The [A] matrix is put into core and subroutine ALLMAT is called.

5. Subroutines Called: CFBSØR, MERGED, CFACTØR, ALLMAT, and HESS2

6. Design Requirements:

1. Open core must be available at /HESS1X/
2. All computations are done single precision.

4.59.8.21 Subroutine Name: HESS2

1. Entry Point: HESS2



## MODULE FUNCTIONAL DESCRIPTIONS

2. Purpose: To form an NRØW X NRØW identity matrix and a proper partitioning vector to insert this identity matrix in a larger matrix.

3. Calling Sequence: CALL HESS2 (NRØW, IDEN, IVP)

where:

NRØW is the order of the identity matrix to be written on IDEN

IDEN - GINØ file number of the Identity matrix

IVP - GINØ file number of the partitioning vector, {IPV} =

$$\left\{ \begin{array}{c} 1.0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{array} \right\} \begin{array}{l} \text{NRØW} \\ \\ \text{NRØW} \end{array} \right\}$$

### 4.59.8.22 Subroutine Name: MERGED

1. Entry Point: MERGED

2. Purpose: To set up and call the matrix utility routine MERGE to perform the following merge

$$\{CP\} \left[ \begin{array}{cc|cc} \xleftarrow{RP} & & & \\ A11 & & A12 & \\ \hline & & & \\ A21 & & A22 & \end{array} \right] \Rightarrow [A]$$

3. Calling Sequence:

CALL MERGED (A11,A12,A21,A22,A,RP,CP,N1,N2)

where:

A11, A12, A21, A22, A, RP, and CP are the GINØ file numbers of their respective data blocks. If any partition does not exist its GINØ file number should be 0. All input data blocks should have matrix trailers. RP or CP may be 0. N1 and N2 are the orders of RP and CP if either is zero.

4. Method: MERGED sets up /PARMEG/ and calls MERGE. It also writes trailer on A.

5. Design Requirements: Open core exists at /MRGEDX/

# FUNCTIONAL MODULE CEAD (COMPLEX EIGENVALUE ANALYSIS - DISPLACEMENT)

## 4.59.8.23 Subroutine Name: CFACTR

1. Entry Point: CFACTR
2. Purpose: CFACTR will decompose a complex matrix A into its two factors LLL and ULL.
3. Calling Sequence: CALL CFACTR (A,LLL,ULL,SCR1,SCR2,SCR3,IØPT)

where:

A, LLL, ULL, SCR1, SCR2, and SCR3 are the GINØ file numbers of their respective data blocks.

IØPT is output according to the following table:

| <u>Decomp Method</u> | <u>IØPT</u> |
|----------------------|-------------|
| CSSP                 | 4           |
| CSDP                 | 2           |
| CUSP                 | 1           |

4. Method: The various decomposition methods are chosen based on /SYSTEM/(55) and the trailer of A. CFACTR will write a trailer on the factors.
5. Subroutines Called: CFACTR may call SCDCMP, CSPSDC, or CDCØMP.
6. Design Requirements: Open core must be at /CFACTX/

## 4.59.8.24 Subroutine Name: CFBSØR

1. Entry Point: CFBSØR
2. Purpose: The purpose of CFBSØR is to solve the complex matrix equation  $[A][X] = [B]$  given the factors of  $[A] = [LLL][ULL]$ .
3. Calling Sequence: CALL CFBSØR (LLL,ULL,B,X,IØPT)

where:

LLL,ULL, B and X are the GINØ file names of their respective matrices.

IØPT is an input parameter set by CFACTR (See Section 4.59.8.24).

4. Method: The trailers of [LLL], [ULL] and [B] are used to prepare for FBS. A trailer is written on [X].
5. Subroutines Called: CXFBS or GFBS may be called based on IØPT.
6. Design Requirements: Open core at /CFBSRX/.



## MODULE FUNCTIONAL DESCRIPTIONS

### 4.59.8.25 Subroutine Name: ALLMAT

1. Entry Point: ALLMAT

2. Purpose: The purpose of ALLMAT is to compute the eigenvalues (All) and eigenvectors (number requested) of an arbitrary complex matrix by use of the QR algorithm and the Wielandt inverse power method for vectors. This is essentially the routine distributed via SHARE as SDA 3441.

3. Calling Sequence: CALL ALLMAT (A,LAMBDA,H,HL,VECT,MULT,INTH,M,NCAL,IØPT1)

where:

- A - is the M X M complex input matrix. On return A contains the complex eigenvectors.
- LAMBDA - is a complex list of eigenvalues.
- H - is a complex M X M working array.
- HL - is a complex M X M working array
- VECT - is a complex M order array
- MULT - is a complex M order array
- INTH and INT are real M order arrays
- M - is the problem order.
- NCAL - is the number of eigenvectors to compute.
- IØPT1 - is not used.

4. Method: The input matrix A is reduced to an upper Hessenberg matrix, H, by a sequence of elementary triangular and permutation matrices which make up a matrix P such that  $P^{-1}AP = H$ . The QR algorithm is made use of in ALLMAT by applying unitary similarity transformations to Hessenberg matrices,  $H_i: H_1 = P^{-1}AP$ .  $H_s = (h_{ij}^{(s)}) = Q_s^H H_s Q_s = Q_s^H Q_s T_s Q_s = T_s Q_s$  where  $Q_s^H$  is the product of plane rotations, chosen so that  $T_s$  is upper triangular. This process makes  $h_{n,n-1}^{(s)}$  converge to zero and therefore  $h_{nn}^{(s)}$  converges to an eigenvalue of A. When convergence is met ( $h_{n,n-1}^{(s)}$  negligible) the Hessenberg matrix,  $H_s$ , is deflated (i.e., last row and column eliminated) and ALLMAT proceeds with its leading principal submatrix (a new  $H_1$ ) of order one less. If  $h_{n-1,n-2}^{(s)}$  becomes negligible, the eigenvalues of the lower right hand matrix of order two are calculated and ALLMAT proceeds with the leading principal matrix of order two less. It can be shown that convergence is accelerated by judiciously subtracting scalar matrices from the  $H_s$  matrices. ALLMAT actually replaces  $H_s$  by  $H_s - k_s I$  such that  $k_s$

## FUNCTIONAL MODULE CEAD (COMPLEX EIGENVALUE ANALYSIS - DISPLACEMENT)

is one of the eigenvalues  $p_s$  or  $q_s$  of the lower right hand  $2 \times 2$  matrix of  $H_s$ . The choice of  $p_s$  or  $q_s$  is made on the basis of whether  $|h_{nn}^{(s)} - p_s|$  or  $|h_{nn}^{(s)} - q_s|$  is a minimum. The shift technique is applied at each iteration.

Two passes of the Wielandt inverse power method are used to calculate the eigenvectors,  $y_i$ , of  $H$ . Very little work is required for the second pass since the necessary elementary triangular and permutation matrices are stored in MULT and INTH (storage areas internal to ALLMAT). Finally, the eigenvectors of  $A$ ,  $Py_i$ , are calculated. The matrix  $P$  resides in INT and the lower part of  $H$  (INT and  $H$  are internal to ALLMAT).

References: The theory and a complete description of the algorithms appear in the first reference.

- (1) J. H. Wilkinson (1965): The Algebraic Eigenvalue Problem, Oxford.
- (2) A. S. Householder (1964): The Theory of Matrices in Numerical Analysis, Blaisdell.

### Authors

R. E. Punderlic, J. Rinzol, Scientific Programming Department, Central Data Processing Facility, Union Carbide Corporation, Nuclear Division, ORGDP, Oak Ridge, Tennessee.

5. Design requirements: The eigenvalues of  $A$  are not necessarily calculated in any absolute algebraic order.

Ten iterations per eigenvalue are allowed and examples exist for which convergence will not occur in ALLMAT (e.g., most lower triangular matrices with all equal eigenvalues; a matrix with ones on the lower diagonal, one as the  $N^{\text{th}}$  component of the first row and zeros elsewhere). In the case of non-convergence, ALLMAT will return a value for NCAL less than  $N$  and it is suggested that the user experiment with arbitrary shifts of the input matrix (i.e., add a constant to the diagonal of  $A$ ) which will sometimes eliminate the difficulty (e.g., second example just stated).

If overflows or detrimental underflows occur, scaling  $A$  such that its largest element is in modulus about one will probably eliminate the difficulty.

The accuracy obtaining in computing the eigenvalues of  $A$ ,  $\lambda_i(A)$ , is usually related to the spectral radius,  $\rho(A)$ , of the matrix  $A$  or more generally to some norm of  $A$  times the norm of its inverse. Hence the greater  $\rho(A)/\min|\lambda_i|$  the fewer significant digits the smaller eigenvalues may have. Accuracy also decreases as the order of the matrix increases. Close eigenvalues are usually calculated with less accuracy than well separated ones. In most cases ALLMAT has yielded roots and vectors accurate to about six significant digits.



## MODULE FUNCTIONAL DESCRIPTIONS

### 4.59.8.26 Subroutine Name: CFCNTL

1. Entry Point: CFCNTL

2. Purpose: CFCNTL is the main driver for the complex version of the Tridiagonal Reduction (FEER) method.

3. Calling Sequence: CALL CFCNTL (EED,METHØD,NFØUND)

EED - GINØ file number for file containing EIGC information (Integer, input).

METHØD - Eigenvalue extraction method 'FEER' (Hollerith, input).

NFØUND - Accumulated number of acceptable eigensolutions (Integer, output).

CØMMØN/FEERAA/IK(7),IM(7),IB(7),ILAM(7),IPHI(7),IDMPFL,ISCR(11),REG(7,10),MCBLT(7),MCBUT(7),  
MCBVEC(7),MCBLMB(7)

IK,IM,IB - Matrix control blocks for the input stiffness, mass and damping matrices,  
respectively.

ILAM,IPHI - Matrix control blocks for the output eigenvalue and eigenvector files.

IDMPFL - GINØ file number for the eigenvalue summary file (integer).

ISCR - GINØ file numbers for eleven scratch files (integer).

REG - Array of information obtained from the EIGC bulk data card and continuation  
cards.

MCBLT - Lower triangular matrix, [L], control block.

MCBUT - Upper triangular matrix, [U], control block.

MCBVEC - Orthogonal vector file control block.

MCBLMB - Matrix control block for  $\lambda_0[M] + [B]$ .

CØMMØN/FEERXC/LAMBDA(2),SYMMET,MREDUC,NØRD,IDIAG,EPS,NØRTHØ,NØRD2,NØRD4,NØRDP1,NSWP,JSKIP,  
NØB,IT,TEN2MT,TENMHT,NSTART,QPR,JREG,NØREG,NZERØ,TENMTT,MINØPN,NUMØRT,NUMRAN

LAMBDA - Point of interest (i.e., center of neighborhood) in the complex plane  
(double precision).

SYMMET - Indicator for symmetric dynamic matrix (logical).

MREDUC - Size of the reduced problem (set internally).

NØRD - Problem size (set internally using the dimension of the stiffness matrix).

IDIAG - DIAG 12 output indicator (set internally).

EPS - The user specified (or default) desired theoretical accuracy of the eigenvalues  
expressed as a percentage (double precision).

# FUNCTIONAL MODULE CEAD (COMPLEX EIGENVALUE ANALYSIS - DISPLACEMENT)

|        |   |
|--------|---|
| NØRTHØ | - Number of orthogonal vectors in present set (includes previously computed vectors).                         |
| NØRD2  | - $2 \times NØRD$ .   |
| NØRD4  | - $4 \times NØRD$ .   |
| NØRDP1 | - $NØRD + 1$ .  |
| NSWP   | - Vector size for sweep algorithm.  |
| JSKIP  | - CFEER4 logic bypass indicator.  |
| NØB    | - Indicator for absence of damping matrix [B] (logical).  |
| IT     | - Number of decimal digits of accuracy, t, for the computer.  |
| TEN2MT | - $10^{2-t}$ , where t is defined by IT; reorthogonalization accuracy criterion.                              |
| TENMHT | - $10^{-t/2}$ , where t is defined by IT; accuracy criterion for disjoint tridiagonal matrix.                 |
| NSTART | - Number of initial reorthogonalization attempts (for the current point of interest).                         |
| QPR    | - Indicator for very detailed printout (logical).   |
| JREG   | - Number (positive integer) of the current neighborhood.  |
| NØREG  | - Total number of neighborhoods, or points of interest, in the complex plane, to be processed.                |
| NZERØ  | - Number of previously obtained eigenvectors.   |
| TENMTT | - $10^{-t/3}$ , where t is defined by IT; rigid body root criterion.  |
| MINØPN | - Minimum open core not used by the complex FEER process, in single precision words.                          |
| NUMØRT | - Total number of reorthogonalizations of all the trial vectors employed.                                     |
| NUMRAN | - Total number of random starting and restart vectors used by the complex FEER process for all neighborhoods. |

CØMMØN/FEERZC/Z(1)

Z - Area of open core used by CFCNTL.

4. Design Requirements: The complex Feer method requires sufficient core for three GINØ buffers and five pairs of (complex) vectors of size NØRD ( $2 \times NØRD$  when NØB=FALSE). The method can be performed in single or double precision.



## MODULE FUNCTIONAL DESCRIPTIONS

### 5. Subroutine Glossary for the complex FEER method.

CFCNTL

CFEER1

CFEER2

CFEER3

CFEER4

CFER3S,CFER3D

CFE1A0,CFE2A0

CFE1MY,CFE2MY

CF1FBS,CF2FBS

CFN0R1,CFN0R2

CF10RT,CF20RT

} Single and double precision versions.

#### 4.59.8.27 Subroutine Name: CFEER1

1. Entry Point: CFEER1

2. Purpose: To set up the call to SADD to form the dynamic matrix

$$[\bar{D}] = \lambda_0^2 [M] + \lambda_0 [B] + [K]$$

and also the sub-factor of the dynamic matrix,

$$[\bar{S}] = \lambda_0 [M] + [B]$$

3. Calling Sequence: CALL CFEER1

COMMON/FEERAA/

COMMON/FEERXC/

See CFCNTL for a description of /FEERAA/ and /FEERXC/ (Section 4.59.8.26)

COMMON/FEERZ1/Z(2)

Z - Area of open core used by CFEER1.

4. Design Requirements: CFEER1 can perform single or double precision operations.

#### 4.59.8.28 Subroutine Name: CFEER2

1. Entry Point: CFEER2

FUNCTIONAL MODULE CEAD (COMPLEX EIGENVALUE ANALYSIS - DISPLACEMENT)

2. Purpose: To set up the call to CDCOMP to decompose the dynamic matrix

$$[\bar{D}] = [L][U]$$

3. Calling Sequence: CALL CFEER2 (IRET)

IRET - Singularity indicator (output).

COMMON/FEERAA/

COMMON/FEERXC/

See CFCNTL for a description of /FEERAA/ and /FEERXC/ (Section 4.59.8.26).

COMMON/FEERZ2/Z(2)

Z - Area of open core used by CFEER2.

4.59.8.29 Subroutine Name: CFEER3

1. Entry Point: CFEER3

2. Purpose: CFEER3 is the basic driver routine which obtains the reduced tridiagonal matrix for the complex FEER method of eigenvalue extraction.

3. Calling Sequence: CALL CFEER3

COMMON/FEERAA/

COMMON/FEERXC/

See CFCNTL for a description of /FEERAA/ and /FEERXC/ (Section 4.59.8.26).

COMMON/FEERZ3/Z(2)

Z - Area of open core used by CFEER3.

4. Design Requirements: CFEER3 can perform single or double precision operations.

4.59.8.30 Subroutine Name: CFEER4

1. Entry Point: CFEER4

2. Purpose: CFEER4 obtains the physical eigenvalues and eigenvectors of the reduced problem (for the current point of interest) by feeding the reduced tridiagonal matrix to subroutine ALLMAT. In addition, CFEER4 sorts the eigenvalues according to increasing distance from the center of interest, and determines which eigensolutions are acceptable and which are not (according to the default or user-input accuracy criterion).



## MODULE FUNCTIONAL DESCRIPTIONS

### 3. Calling Sequence: CALL CFEER4

~~COMMON~~/FEERAA/

~~COMMON~~/FEERXC/

See CFCNTL for a description of /FEERAA/ and /FEERXC/ (Section 4.59.8.26).

~~COMMON~~/FEERZ4/Z(2)

Z                - Area of open core used by CFEER4.

### 4. Design Requirements: CFEER4 performs only single precision operations.

#### 4.59.8.31 Subroutines CFER3S,CFER3D

##### 1. Entry Points: CFER3S (single precision), CFER3D (double precision).

##### 2. Purpose: To perform the tridiagonal reduction algorithm for CFEER3.

##### 3. Calling Sequence: CALL CFER3S(V1,V1L,V2,V2L,V3,V3L,V4,V4L,V5,V5L,ZB,ZC)

CALL CFER3D (same arguments as for CFER3s).

|        |   |   |
|--------|---|---|
| V1,V1L | } | - Working space for five pairs of vectors, where a vector pair consists of a right-hand vector (e.g., V1) and a left-hand vector (e.g., V1L). |
| V2,V2L |   |   |
| V3,V3L |   |   |
| V4,V4L |   |   |
| V5,V5L |   |   |

ZB,ZC            - Working space for two GINØ buffers

~~COMMON~~/FEERAA/

~~COMMON~~/FEERXC/

See CFCNTL for a description of /FEERAA/ and /FEERXC/ (Section 4.59.8.26).

4. Design Requirements: CFER3S and CFER3D are driven by the CFEER3 subprogram, which performs the necessary initialization and termination process. CFEER3 invokes either CFER3S or CFER3D according to the precision of the required input files. All vectors are dimensioned consistent with the initial problem size (NORD2 when NOB=TRUE and NORD4 otherwise). In addition, each left-hand vector must immediately follow its corresponding right-hand vector in core, since this configuration results in more streamlined coding and associated quicker execution.

#### 4.59.8.32 Subroutines CFE1AØ,CFE2AØ

##### 1. Entry Point: CFE1AØ (single precision), CFE2AØ (double precision).

FUNCTIONAL MODULE CEAD (COMPLEX EIGENVALUE ANALYSIS - DISPLACEMENT)

2. Purpose: To perform the eigenmatrix multiplication operation.

3. Calling Sequence: CALL  $\begin{matrix} \text{CFE1A0} \\ \text{CFE2A0} \end{matrix}$  (TP0SE,V1,V2,V3,ZB) } Single and double  
precision versions

TP0SE - Indicator for transpose operation (logical).

V1 - Input vector.

V2 - Output vector.

V3 - Working space for one vector.

ZB - Working space for one GIN0 buffer.

C0MM0N/FEERAA/

C0MM0N/FEERXC/

See CFCNTL for a description /FEERAA/ and /FEERXC/ (Section 4.59.8.26)

4. Design Requirements: All vectors are of dimension consistent with the initial problem size (see Section 4.59.8.31).

4.59.8.33 Subroutines CFE1MY,CFE2MY

1. Entry Points: CFE1MY (single precision), CFE2MY (double precision).

2. Purpose: To perform the standard matrix-times-vector multiply function. The operation is

$$\{X\} = [M]\{Y\}$$

or

$$\{X\} = [M]^T\{Y\}$$

3. Calling Sequence: CALL  $\begin{matrix} \text{CFE1MY} \\ \text{CFE2MY} \end{matrix}$  (TP0SE,Y,X,FILE,BUF).



## MODULE FUNCTIONAL DESCRIPTIONS

- TPØSE - Indicator for transpose operation (logical).
- Y - Input vector
- X - Output vector.
- FILE(7) - Input matrix control block for the required matrix (integer).
- BUF - Working space for one GINØ buffer.

4. Design Requirements: All vectors are of dimension consistent with the initial problem size.

### 4.59.8.34 Subroutines CF1FBS,CF2FBS

1. Entry Point: CF1FBS (single precision), CF2FBS (double precision).
2. Purpose: To perform the operational inverse algorithm (forward and backward sweeps).  
The operation is

$$\{X\} \leftarrow [L][U] \{X\}$$

or

$$\{X\} \leftarrow [U]^T [L]^T \{X\}$$

3. Calling Sequence: CALL CF1FBS (TPØSE,XØUT,IØBUF)      Single and double  
CF2FBS      precision versions

- TPØSE - Indicator for transpose operation (logical).
- XØUT - Input vector, which gets transformed to the output vector.
- IØBUF - Working space for one GINØ buffer.

CØMMØN/FEERAA/

CØMMØN/FEERXC/

See CFCNTL for a description of /FEERAA/ and /FEERXC/ (Section 4.59.8.26).

4. Design Requirments: CF1FBS and CF2FBS use the direct output of the CDCØMP subprogram.  
The vector is dimensioned consistent with half of the initial problem size when the damping matrix is present, and consistent with the initial problem size when the damping matrix is absent.

### 4.59.8.35 Subroutines CFNØR1,CFNØR2

1. Entry Point: CFNØR1 (single precision), CFNØR2 (double precision).

# FUNCTIONAL MODULE CEAD (COMPLEX EIGENVALUE ANALYSIS - DISPLACEMENT)

2. Purpose: To normalize a pair of (right-hand and left-hand) complex vectors to magnitude unity. The operation is

$$\beta = [\{\bar{W}\}^T \{W\}]^{1/2}$$

$$\{V\} = \frac{1}{\beta} \{W\}$$

$$\{V\} = \frac{1}{\beta} \{\bar{W}\}$$

where the bar denotes a left-hand vector.

3. Calling Sequence: CALL  $\begin{matrix} \text{CFNØR1} \\ \text{CFNØR2} \end{matrix}$  (RIGHT,LEFT,SIZE2,ØPTIØN,RI) } Single and double precision versions

- RIGHT - Input right-hand complex vector, which gets transformed to the output right-hand vector when ØPTIØN = 0.
- LEFT - Input left-hand complex vector, which gets transformed to the output left-hand vector when ØPTIØN = 0.
- SIZE2 - Length of either vector in computer words (integer).
- ØPTIØN - Selects the desired option, as follows:
  - (0) Normalize the input vectors, and output the square root of the inner product in RI.
  - (1) Only output the inner product, in RI.
  - (2) Only output the square root of the inner product, in RI.
- RI - Inner product, or square root of the inner product, or the input factors (output; see ØPTIØN).

CØMMØN/FEERCX/

See CFCNTL for a description of /FEERCX/ (Section 4.59.8.26).

## 4.59.8.36 Subroutines CF1ØRT,CF2ØRT

- 1. Entry Point: CF1ØRT (single precision), CF2ØRT (double precision)
- 2. Purpose: To perform the reorthogonalization algorithm.
- 3. Calling Sequence: CALL  $\begin{matrix} \text{CF1ØRT} \\ \text{CF2ØRT} \end{matrix}$  (SUCESS,MAXITS,TEN2MT,NZERØ,IØRTHØ,VR,VL,VL,V1L,V2,V2L,ZB)
- SUCESS - Output indicator for successful reorthogonalization (logical).
- MAXITS - Input maximum allowed number of reorthogonalization iterations (integer).



## MODULE FUNCTIONAL DESCRIPTIONS

- TEN2MT        - Convergence tolerance (input).
- NZERØ        - Number of orthogonal vector-pairs (right and left hand) from restart and  
prior neighborhoods (input).
- IØRTHØ       - Number of orthogonal vector-pairs previously computed in the current  
neighborhood (input).
- VR,VL        - Input (and transformed for output) right and left hand vectors, respectively,  
which are to be reorthogonalized with respect to all pairs of previously com-  
puted orthogonal vectors.
- V1,V1L  
V2,V2L       - Working space for four vectors.
- ZB            - Working space for one GINØ buffer.

CØMMØN/FEERAA/

CØMMØN/FEERXC/

See CFCNTL for a description of /FEERAA/ and /FEERXC/ (Section 4.59.8.26).

4. Design Requirements: All vectors are dimensioned consistent with the initial problem size (see Section 4.59.8.31). In addition, V1L must immediately follow V1 in core.

### 4.59.8.37 Subroutine Name: CLVEC

1. Entry Point: CLVEC
2. Purpose: To generate the left eigenvectors for the Determinant and Hessenburg Methods if requested. (Note: The left eigenvectors are already generated during a normal Inverse Power approach.)

3. Calling Sequence: CALL CLVEC (LAMD,NVECT,PHIDL,IH,IBUF,IBUF1)

LAMD - GINØ file number of sorted eigenvalues - integer-input.

NVECT - Number of left eigenvectors to be calculated - integer-input.

PHIDL - GINØ file number of data block PHIDL - integer-input.

IH - Trailer for data block PHIDL - integer array-input,output.

IBUF - Open core pointer to GINØ buffer - integer-input.

IBUF1 - Open core pointer to GINØ buffer - integer-input.

4. Method: For each eigenvalue,  $\lambda_1$ , a dynamic matrix of the form  $M\lambda_1^2 + B\lambda_1 + K$  is formed and decomposed into its upper and lower triangular components U and L. An arbitrary load

#### FUNCTIONAL MODULE CEAD (COMPLEX EIGENVALUE ANALYSIS - DISPLACEMENT)

vector  $F$  is generated and the equation  $U_L^t \phi_i = F$  is solved for the left eigenvector  $\phi_i$ .  $\phi_i$  is normalized and packed into output file PHIDL. The process is repeated for the first NVECT eigenvalues on file LAMD. Inverse Power subroutines CINVPI, CINFP2, CDIFBS and CNØRM1 are used in performing the above operations.

##### 4.95.9 Design Requirements

Open core is defined at /CEAD1X/ to process EED. Open core is defined at /CEAD1A/ for use by CEAD1A.

##### 4.59.10 Diagnostic Messages

The following diagnostic messages may be generated: 3001, 3002, 3003, 3005, 3007, 3008, 3025 and 3045.



#### 4.60 FUNCTIONAL MODULE VDR (VECTOR DATA RECOVERY)

##### 4.60.1 Entry Point: VDR

##### 4.60.2 Purpose

VDR formats data blocks for input to the Output File Processor (ØFP) and XY plot (XYPLØT) modules to provide a capability for output of vectors in the solution set.

##### 4.60.3 DMAP Calling Sequence

$$\text{VDR} \left\{ \begin{array}{l} \text{CASECC} \\ \text{CASEXX} \end{array} \right\}, \left\{ \begin{array}{l} \text{EQDYN} \\ \text{HEQDYN} \end{array} \right\}, \left\{ \begin{array}{l} \text{USETD} \\ \text{HUSETD} \end{array} \right\}, \left\{ \begin{array}{l} \text{PHID} \\ \text{UDVF} \\ \text{UDVT} \\ \text{PHIH} \\ \text{UHVT} \\ \text{HUDVT} \end{array} \right\}, \left\{ \begin{array}{l} \text{CLAMA} \\ \text{PPF} \\ \text{TØL} \\ \text{HTØL} \end{array} \right\}, \text{XYCDB}, \left\{ \begin{array}{l} \text{PNLD} \\ \text{PNLH} \\ \text{HPNLD} \end{array} \right\}, \left\{ \begin{array}{l} \text{ØPHID} \\ \text{ØUDVC1} \\ \text{ØUDV1} \\ \text{ØPHIH} \\ \text{ØUHVC1} \\ \text{ØUHV1} \\ \text{HØUDV1} \end{array} \right\}, \left\{ \begin{array}{l} \text{ØPNL1} \\ \text{HØPNL1} \end{array} \right\} /$$

$$\text{C,N}, \left\{ \begin{array}{l} \text{TRANRESP} \\ \text{FREQRESP} \\ \text{CEIGN} \end{array} \right\} / \text{C,N}, \left\{ \begin{array}{l} \text{DIRECT} \\ \text{MØDAL} \end{array} \right\} / \text{V,N}, \text{SØRT2/V,N}, \text{ØUTPUT/V,N}, \text{SDR2/V,N}, \text{FMØDE \$}$$

##### 4.60.4 Input Data Blocks

CASECC } - Case Control Data Table.  
CASEXX }

EQDYN } - Equivalence between external and internal number - Dynamics.  
HEQDYN }

USETD - Displacement set definitions table - Dynamics.

PHID } - Partition of displacement vector.  
UDVF }  
UDVT }  
PHIH }  
UHVT }

CLAMA - Complex eigenvalue, table.

PPF - Dynamic loads for frequency response - p set.

TØL } - Table of output times.  
HTØL }

XYCDB - X Y Control Data Block

PNLD } - Non-Linear Load Vector.  
PNLH }  
HPNLD }

HUSETD - Temperature set definitions table - Dynamics.

HUDVT - Partition of temperature vector.

## MODULE FUNCTIONAL DESCRIPTIONS

### Notes:

1. CASECC, EQDYN and USETD may not be purged.
2. PP may be purged only if UDV is purged.
3. PNL and XYCDB may be purged.

### 4.60.5 Output Data Blocks

$\left. \begin{array}{l} \emptyset PHID \\ \emptyset UDVC1 \\ \emptyset UDV1 \\ \emptyset PHIH \\ \emptyset UHVC1 \\ \emptyset UHVI \end{array} \right\} - \text{Output displacement requests - Solution set.}$

$\left. \begin{array}{l} \emptyset PNL1 \\ H\emptyset PNL1 \end{array} \right\} - \text{Non-Linear Load Vector.}$

$H\emptyset UDV1 - \text{Output temperature requests - Solution set.}$

Note: Output data blocks may be purged.

### 4.60.6 Parameters

The first parameter indicates a Rigid Format and must be one of the three names shown above. The second parameter indicates a direct or modal formulation and must be one of the two names shown above.

$S\emptyset RT2 - \text{Output-integer-no default. +1 if any } S\emptyset RT2 \text{ output is requested, -1 otherwise.}$

$\emptyset UTPUT - \text{Output-integer-no default. +1 if any output in the solution set is requested, -1 otherwise.}$

$SDR2 - \text{Output-integer-no default. +1 if any requests for output in the physical set are found in CASECC or XYCDB, -1 otherwise.}$

$FM\emptyset DE - \text{Input-integer-no default. If a modal formulation, } FM\emptyset DE = \text{mode number of the first mode. } FM\emptyset DE \text{ is not used in a direct formulation.}$



## FUNCTIONAL MODULE VDR (VECTOR DATA RECOVERY)

### 4.60.7 Method

#### 4.60.7.1 General

VDR is the main control program for the module. VDRA is called to analyze the Case Control (CASECC) and XYCDB data blocks. If any requests for solution set output are found, VDRB is called to assemble the ØUDV1 output data block for processing by the ØFP. If the problem is a transient response problem, VDRB is called a second time to process any requests for non-linear load output.

#### 4.60.7.2 Analysis of the Case Control and XYCDB Data Blocks

VDRA attempts to open the XYCDB data block. If it is purged, a return is given to VDR. Otherwise, the header record and first data record of XYCDB are skipped, and data applying to all subcases are read from the second data record. If no such data exist, a dummy master is created. Otherwise, the master data are reduced to a list of unique pairs. If only master data exist, flags are set appropriately.

For each record in the Case Control data block the following processing occurs:

1. The record is read into core. If no XYCDB subcase corresponds to the Case Control subcase, pointers are set to the master data. Otherwise, the master data and appropriate XYCDB subcase data are merged and reduced to unique pairs.
2. For each request for solution set output in XYCDB, the corresponding request in Case Control is examined. If no request is present in Case Control, the XYCDB request is reduced to a set in Case Control format, and a request for the set is turned on in Case Control. If the Case Control set is "ALL", no further action is taken. If the Case Control request is a set, the set is merged with the XYCDB set, and the request altered to reflect the new set (unless all points in the XYCDB set were already in the Case Control set). A flag is set if any new requests are formed.
3. When all requests for the current Case Control record have been analyzed, the record (as modified) is written on a scratch file.
4. When all Case Control records have been read, the GINØ file name for the Case Control data block is switched to the scratch file (unless no modifications were made to Case Control).

## MODULE FUNCTIONAL DESCRIPTIONS

### 4.60.7.3 Preparation of Solution Set Output

The operations of VDRB are dependent on the Rigid Format being executed. VDRB operates in all six of the dynamics Rigid Formats. The initial operations in VDRB proceed as follows:

1. For a direct solution, or a modal solution with extra points, the second record of EQDYN is read into core. USETD is read into core.
2. If the problem is a direct solution, each entry in EQDYN is processed. The scalar index value (the 2nd word of each entry) is replaced by the scalar index value in the solution set plus a code indicating which components of the point are in the solution set.
3. If the problem is a modal solution with extra points, the scalar index of each extra point in EQDYN is replaced with a scalar index in the solution set. The scalar indices of all other points are replaced with zero.
4. If the problem is a complex eigenvalue problem, a list of mode numbers and complex eigenvalues is read into core from the CLAMA data block.
5. If the problem is a transient response problem, a list of times is read into core from the TØL data block.
6. If the problem is a frequency response problem, a list of frequencies is read into core from the PP data block.
7. The header record on the input file is skipped, and various parameters are initialized for the overall processing.

A record on the Case Control data block is read. The output request is examined. If the output is defined in terms of a set, pointers to the set definition are computed. The vector is unpacked in core (unless the vector is already in core in the case of velocities and accelerations for frequency problems).

Information is assembled to write the identification record on the output data block as follows.

1. For complex eigenvalues, the mode number and eigenvalue are picked up from the list in core.



## FUNCTIONAL MODULE VDR (VECTOR DATA RECOVERY)

2. For frequency response, the frequency is picked up from the list in core. A comparison with the ØFREQ selection in Case Control is made. If the current frequency is not marked for output, the remainder of the calculations for the current vector are skipped.

3. For a transient problem, the time is picked up.

The identification record is written. Entries are written in the data record according to the request. The modified EQDYN table in core is used to pick up points in the vector to be output. Conversion to magnitude and phase is made if requested.

When all points in the current request have been processed, post processing occurs depending on the problem type as follows:

1. For complex eigenvalues, a pointer is updated to the next mode number and eigenvalue. If all eigenvectors have not been processed, the steps above are repeated. Otherwise, terminal processing is initiated.
2. For frequency response, if the vector just processed was a displacement vector, the corresponding velocity vector is determined by differentiating with respect to time.

$$\{v\} = i\omega \{u\}. \quad (1)$$

Similarly, if the vector just processed was a velocity vector, the corresponding acceleration vector is formed by differentiating with respect to time:

$$\{a\} = i\omega \{v\}. \quad (2)$$

If all vectors have not been processed, the steps above are repeated. Otherwise, terminal processing is initiated.

3. For transient response, pointers are updated so that the vectors will be processed in the order a) displacement, b) velocity, and c) acceleration. If all vectors have not been processed, the steps above are repeated. Otherwise, terminal processing is initiated.

The terminal processing consists of closing all files, writing a trailer on the output file and exiting.



## MODULE FUNCTIONAL DESCRIPTIONS

### 4.60.8 Subroutines

#### 4.60.8.1 Subroutine Name: VDR

1. Entry Point: VDR
2. Purpose: Main control program for the module.
3. Calling Sequence: CALL VDR

#### 4.60.8.2 Subroutine Name: VDRA

1. Entry Point: VDRA
2. Purpose: To analyze the output requests in the Case Control and XYCDB data blocks.
3. Calling Sequence: CALL VDRA

#### 4.60.8.3 Subroutine Name: VDRB

1. Entry Point: VDRB
2. Purpose: To process requests for solution set output and assemble the output data block.
3. Calling Sequence: CALL VDRB (INFIL,ØUTFL,IREQ)

INFIL - GINØ file name of the data block containing vectors to be output in the solution set.

ØUTFL - GINØ file name of the data block where solution set output will be written.

IREQ - Word position in the Case Control record where solution set output request is defined.

### 4.60.9 Design Requirements

#### 4.60.9.1 Allocation of Core Storage

## FUNCTIONAL MODULE VDR (VECTOR DATA RECOVERY)

The maximum storage requirements for the module are in VDRB. A general picture of core storage is as follows:

| COMMON/VDRCDR/Z(1) |   |  |
|--------------------|---|--|
| 1                  | EQDYN Table                               | } 2 words per entry, one entry for each point in the problem.                      |
| ILIST              | List of eigenvalues, frequencies or times |  |
| ICC+1              | Case Control record                       | } 1, 2 or 3 words per entry, one entry for each eigenvalue, frequency or time.     |
| IVC                | Unpacked Vector                           |  |
| BUF3               | Buffer for input file                     | } One word for each degree of freedom in the solution set. (two words if complex). |
| BUF2               | Buffer for output file                    |  |
| BUF1               | Buffer for Case Control                   |  |

### 4.60.9.2 Environment

The Block Data program VDRBD initializes /VDRCDM/ with GINØ file names, data defining position of parameters in a Case Control record, data defining rigid formats and problem types, and miscellaneous data. It must be in core when VDR is executed.

The module VDR is designed to be executed as one overlay segment. Open core is defined by /VDRCDR/. Two scratch files are used.

#### 4.61 FUNCTIONAL MODULE FRRD (FREQUENCY RESPONSE - DISPLACEMENT APPROACH)

##### 4.61.1 Entry Point: FRRD

##### 4.61.2 Purpose

To solve the matrix equation

$$[-\omega_i^2 [M] + i\omega [B] + [K]] [X] = [P(\omega_i)]$$

at a given set of frequencies  $\omega_i$  and loads P (which may be functions of  $\omega_i$ ).

##### 4.61.3 DMAP Calling Sequence

FRRD CASECC, USETD, DLT, FRL, GMD, GØD,  $\left\{ \begin{matrix} KDD \\ KHH \end{matrix} \right\}$ ,  $\left\{ \begin{matrix} BDD \\ BHH \end{matrix} \right\}$ ,  $\left\{ \begin{matrix} MDD \\ MHH \end{matrix} \right\}$ , PHIDH, DIT/UHV, PS, PD, PP/V, N, APP/  
V, N, FØRM/V, N, LUSETD/V, N, MPCF1/V, N, SINGLE/V, N, ØMIT/V, N, NONCUP/V, N, FRQSET/C, Y, DECØMØPT=1 \$

##### 4.61.4 Input Data Blocks

CASECC - Case Control Data table.  
USETD - Displacement set definitions table dynamics.  
DLT - Dynamic Loads Table.  
FRL - Frequency Response List.  
GMD - Multipoint constraint transformation matrix - dynamics.  
GØD - Omitted coordinate transformation matrix - dynamics.  
KDD - Modal stiffness matrix - d set.  
KHH - Modal stiffness matrix - h set.  
BDD - Modal damping matrix - d set.  
BHH - Modal damping matrix - h set.  
MDD - Modal mass matrix - d set.  
MHH - Modal mass matrix - h set.  
PHIDH - Transformation matrix from d set to modal coordinates.  
DIT - Direct Input Tables.

Notes: 1. CASECC cannot be purged.  
2. USETD cannot be purged.  
3. DLT cannot be purged.  
4. FRL cannot be purged.  
5. GMD cannot be purged if MPCF1  $\geq$  0.  
6. GØD cannot be purged if ØMIT  $\geq$  0.



## MODULE FUNCTIONAL DESCRIPTIONS

7. PHIDH cannot be purged if FØRM = MØDAL.
8. DIT cannot be purged if a load uses tables.

### 4.61.5 Output Data Blocks

- UHV - Displacement vectors.
- PS - Partition of load vector matrix giving loads in s set.
- PD - Load vectors - d set.
- PP - Load vectors - p set.

Notes: 1. UHV, PD, and PP cannot be purged.  
2. PS cannot be purged if SINGLE  $\geq 0$ .

### 4.61.6 Parameters

- APP - Input-BCD-no default. APP should be set equal to DISP.
- FØRM - Input-BCD-no default. FØRM = MØDAL implies a modal solution should be used.
- LUSETD - Input-integer-no default. LUSETD indicates length of p set.
- MPCF1 - Input-integer-no default. MPCF1  $\geq 0$  implies multipoint constraints present.
- SINGLE - Input-integer-no default. SINGLE  $\geq 0$  implies single-point constraints present.
- ØMIT - Input-integer-no default. ØMIT  $\geq 0$  implies omitted coordinates present.
- NØNCUP - Input-integer-no default. NØNCUP = -1 implies noncoupled solution if FØRM = MØDAL.
- FRQSET - Output-integer-no default. FRQSET is the set id of the selected frequency list from CASECC.

### 4.61.7 Method

#### 4.61.7.1 Overview of the Method

The Frequency Response module for the displacement approach assembles a frequency-dependent load vector and solves for the steady-state, frequency response, displacement vectors. Various load sets are defined as functions of frequency. Combinations of these sets are used with the various specified frequencies. Load vectors for each frequency are formed and reduced to loads on the proper degree of freedom. The solutions for both direct formulation and coupled modal formulation are identical except that different matrices are used. The solution involves a triangular decomposition and back substitution using the type of arithmetic selected by the matrix types for each frequency according to the following table.

# FUNCTIONAL MODULE FRRD (FREQUENCY RESPONSE - DISPLACEMENT APPROACH)

| <u>Trailer</u> | <u>Complex Terms</u> | <u>Decomp Method</u> |              |
|----------------|----------------------|----------------------|--------------|
|                |                      | <u>CDC</u>           | <u>Other</u> |
| SYM            | No                   | RSSP                 | RSDP         |
| SYM            | Yes                  | CSSP                 | CSDP         |
| UNSYM          | No                   | RUDP                 | RUDP         |
| UNSYM          | Yes                  | CUDP                 | CUDP         |

where:

SYM = symmetric dynamic matrix  
 UNSYM = unsymmetric dynamic matrix  
 RSSP = real symmetric single precision  
 RSDP = real symmetric double precision  
 RUDP = real unsymmetric double precision  
 CUDP = complex unsymmetric double precision  
 CSSP = complex symmetric single precision  
 CSDP = complex symmetric double precision

The solutions for the uncoupled modal formulation are analytic equations.

## 4.61.7.2 Logical Phases

1. The load vectors for each desired frequency are assembled from the DLT data block. The DLØAD section of the DLT tells which load sets to use and what scale factors to use in combining the load sets. The data for each load set are given in the RLØAD or TLØAD section of the DLT. This is done in subroutine FRRD1A.
2. The total load vectors are partitioned and manipulated to produce load vectors on the solution coordinates. This is done in subroutine FRRD1B.
3. The matrix equation for displacements is now solved for each load combination and each frequency. The overall dynamic matrix is formed. The matrix is decomposed, and the displacements are formed by back substitution using the various loads. If the formulation is an uncoupled modal system, the displacements are calculated directly. This is done by subroutines FRRD1C and FRRD1D or FRRD1F.



## MODULE FUNCTIONAL DESCRIPTIONS

4. The solution vectors are then resorted into load-frequency order. This work is done by subroutine FRRDIE.

### 4.61.7.3 Algorithms

#### 1. Assembly of Load Vectors:

The frequency set id is extracted from CASECC. This frequency set is placed in core from the FRL and converted from radians to frequency. These frequencies are output into the header of PPF for later output identification. The load id is read from CASECC, found in DLT, and a table is constructed giving a simple id and a scale factor for each component. The DLT data are read for each simple id, and a list of the required tables is extracted. Core is allocated to hold as many load vectors as possible up to the number of frequencies. If tables are present, they are initialized and evaluated for all frequencies in core. The DLT is read, and two types of loads are constructed:

$$1) \quad \text{RLØAD1} \quad P(f) = A[C(f) + iD(f)]e^{i(\theta - 2\pi f\tau)} \quad (2)$$

$$2) \quad \text{RLØAD2} \quad P(f) = AB(f)e^{i(\phi(f) + \theta - 2\pi f\tau)} \quad (3)$$

where A, B, C, D,  $\phi$ ,  $\theta$  and  $\tau$  are user input constants or tables.

TLØAD loads are computed as follows:

FUNCTIONAL MODULE FRRD (FREQUENCY RESPONSE - DISPLACEMENT APPROACH)

Case 1. TLØAD1 data card, referencing a TABLED1, 2 or 3.

The P(t) is given in terms defined on the above data cards by

$$P_j(t) = A_j Y_T \left( \frac{t - \tau_j - X1}{X2} \right) \quad (1a)$$

$Y_T$  is a piecewise linear table, for the (N-1) intervals  $(x_1, x_2) \dots (x_{N-1}, x_N)$ . Then,

$$P_j(\omega) = A_j e^{-i\omega\tau_j} X2 \sum_{i=1}^{N-1} \frac{x_{i+1} - x_i}{2} (L_i Y_i + R_i Y_{i+1}) \quad (1b)$$

where

$$L_i = e^{-i\omega(X1+X2 \cdot x_i)} E_2(-i\omega X2(x_{i+1} - x_i)) \quad (1c)$$

$$R_i = e^{-i\omega(X1+X2 \cdot x_{i+1})} E_2(i\omega X2(x_{i+1} - x_i)) \quad (1d)$$

$E_2(i\theta)$  is computed by special formulas for large and small  $\theta$ , as defined in module IFT.

$E_2(-i\theta)$  is the conjugate of  $E_2(\theta)$ .

Case 2. TLØAD1 with TABLED4.

$$P_j(t) = \begin{cases} A_j \sum_{n=0}^N a_n \left( \frac{t - \tau_j - X1}{X2} \right)^n & X3 < t - \tau_j < X4 \\ 0 & \text{Otherwise} \end{cases} \quad (2a)$$

Then,

$$\tilde{P}_j(\omega) = A_j e^{-i\omega\tau_j} (\text{SUM}) \quad (2b)$$

$$(\text{SUM}) = e^{-i\omega X4} (X4 - X3) \sum_{n=0}^N \frac{\tilde{a}_n}{n+1} E_{n+1}(i\omega(X4 - X3)) \quad (2c)$$

$$\tilde{a}_n = \left( \frac{X4 - X3}{X2} \right)^n \sum_{m=0}^{N-n} \frac{(n+m)!}{n!m!} \left( \frac{X3 - X1}{X2} \right)^m a_{n+m} \quad (2d)$$



# MODULE FUNCTIONAL DESCRIPTIONS

The function  $E_n(i\theta)$  is computed as follows:

If  $0 < \theta < 0.1$ , compute for  $n=N$

$$\operatorname{Re} E_N(i\theta) = 1 - \frac{\theta^2}{(N+1)(N+2)} + \frac{\theta^4}{(N+1)\dots(N+4)} - + \dots \quad (2e)$$

$$\operatorname{Im} E_N(i\theta) = \frac{\theta}{N+1} - \frac{\theta^3}{(N+1)\dots(N+3)} + - \dots \quad (2f)$$

Stop when last term  $< 10^{-9}$ .

Then, by recursion for  $n=N-1, N-2, \dots, 1$

$$\operatorname{Re} E_n(i\theta) = 1 - \frac{\theta}{n+1} \operatorname{Im} E_{n+1}(i\theta) \quad (2g)$$

$$\operatorname{Im} E_n(i\theta) = \frac{\theta}{n+1} \operatorname{Re} E_{n+1}(i\theta) \quad (2h)$$

If  $0.1 < \theta < \infty$ , compute for  $n=0$

$$\operatorname{Re} E_0(i\theta) = \cos \theta \quad (2i)$$

$$\operatorname{Im} E_0(i\theta) = \sin \theta \quad (2j)$$

Then, by recursion for  $n=1, 2, 3, \dots, N$

$$\operatorname{Re} E_n(i\theta) = \frac{n}{\theta} \operatorname{Im} E_{n-1}(i\theta) \quad (2k)$$

$$\operatorname{Im} E_n(i\theta) = \frac{n}{\theta} [1 - \operatorname{Re} E_{n-1}(i\theta)] \quad (2l)$$

Case 3. TLØAD2 with

$$P_j(t) = \begin{cases} A_j \tilde{t}^B e^{c\tilde{t}} \cos(2\pi F\tilde{t} + P) & 0 < \tilde{t} < T_2 - T_1 \\ 0 & \text{Otherwise} \end{cases} \quad (3a)$$

$$\tilde{t} \equiv t - T_1 - \tau_j$$

B will be restricted to an integer  $\geq 0$ .

$$\tilde{P}_j(\omega) = A_j e^{-i\omega\tau_j} [R_2 + R_1] \left[ \frac{(T_2 - T_1)^{B+1}}{2(B+1)} \right] \quad (3b)$$

FUNCTIONAL MODULE FRRD (FREQUENCY RESPONSE - DISPLACEMENT APPROACH)

$$R_2 = e^{\text{power}} E_{B+1}(z) \quad (3c)$$

where  $\text{power} = +iP + (c + i2\pi F)(T_2 - T_1) - i\omega T_2$

and  $z = -[c + i2\pi F - i\omega](T_2 - T_1)$

$R_1$  = same as  $R_2$  except the sign of  $P$  and  $F$  are reversed.

$$E_B(z) = \begin{cases} 1 + \frac{z}{B+1} + \frac{z^2}{(B+1)(B+2)} + \dots & |z| < .1 \\ \text{(until last term} < 10^{-9} \text{)} & \\ \frac{B!}{z^B} \left( e^z - \sum_{k=0}^{B-1} \frac{z^k}{k!} \right) & |z| \geq .1 \end{cases} \quad (3d)$$

If all frequencies cannot be evaluated at once, additional passes through the DLT are made until all are evaluated. If additional subcases exist in CASECC, the above steps are repeated for each load.



## MODULE FUNCTIONAL DESCRIPTIONS

### 2. Manipulation of Load Vectors:

The vectors produced in the previous sections are related to the p set. They are reduced by the following steps using data blocks USETD, GMD and GØD.

If MPCF1  $\geq$  0:

$$\{P_p\} \Rightarrow \left\{ \frac{\bar{P}_{ne}}{P_m} \right\}, \quad (4)$$

$$\{P_{ne}\} = \{\bar{P}_{ne}\} + [G_m^d]^T \{P_m\}. \quad (5)$$

If SINGLE  $\geq$  0:

$$\{P_{ne}\} \Rightarrow \left\{ \frac{P_{fe}}{P_s} \right\}. \quad (6)$$

$\{P_s\}$  is output on data block PS.

If ØMIT  $\geq$  0:

$$\{P_{fe}\} \Rightarrow \left\{ \frac{\bar{P}_d}{P_o} \right\}, \quad (7)$$

$$\{P_d\} = \{\bar{P}_d\} + [G_o^d]^T \{P_o\}, \quad (8)$$

$\{P_d\}$  is output on PD.

If FØRM = MØDAL:

$$\{P_h\} = [\Phi_{dh}]^T \{P_d\}. \quad (9)$$

### 3. Solution Phase:

For a direct formulation the equation to be solved is:

$$[-\omega^2 [M_{dd}] + i\omega [B_{dd}] + [K_{dd}]] \{u_d\} = \{P_d(\omega)\}. \quad (10)$$

For a coupled formulation the equations to be solved is:

$$[-\omega^2 [M_{hh}] + i\omega [B_{hh}] + [K_{hh}]] \{u_h\} = \{P_n(\omega)\}. \quad (11)$$

## FUNCTIONAL MODULE FRRD (FREQUENCY RESPONSE - DISPLACEMENT APPROACH)

The left hand matrix is generated by two calls to ADD and decomposed. The normal matrix decomposition checks are relaxed in these solutions. It is expected that the matrices will not pass the triangular decomposition at certain frequencies. The solution will proceed, and only a warning will be issued. The loads at the given frequency are collected from the load file and fed to GFBS for a forward backward substitution solution. If the decomposition failed, a zero vector will result.

For one uncoupled modal formulation the equations to be solved are:

$$\{\epsilon_i\} = \frac{P_i(\omega)}{-m_i\omega^2 + ib_i\omega + k_i} \quad (12)$$

With zero damping the uncoupled modal formulation may produce division by small numbers. This fact is noted and the solution proceeds.

### 4. Order Phase:

Except for the uncoupled modal approach it may be necessary to reorder the solutions from a frequency / load sort to a load / frequency sort.

#### 4.61.8 Subroutines

Utility subroutines PRETAB,TAB,CALCV,SSG2B,SSG2A,SSG2C,CDCØMP,SCDCMP,CSPSDC,CXFBS,FACTØR, SDCØMP, DECØMP, SSG3A and GFBS are used. See subroutine descriptions, Section 3, for details.

##### 4.61.8.1 Subroutine Name: FRRD1A

1. Entry Point: FRRD1A
  2. Purpose: To assemble the user selected loads.
  3. Calling Sequence: CALL FRRD1A (DLT,FRL,CASECC,DIT,PP,LUSETD,NFREQ,NLØAD,FRQSET,FØL)
- DLT,FRL,CASECC,DIT,PP are GINØ file numbers of their respective data blocks - integer - input.
- LUSETD - Length of p set - integer - input.
- NFREQ - Number of frequencies in selected frequency set - integer - output.
- FØL - the GINØ file number of the output frequency list - may be purged.



## MODULE FUNCTIONAL DESCRIPTIONS

NLOAD - Number of loads (records in CASECC) selected - integer - output.

FRQSET - Set id of selected frequency set - integer - output.

### 4.61.8.2 Subroutine Name: FRRD1B

1. Entry Point: FRRD1B

2. Purpose: To reduce loads from the p to the d (or h) set.

3. Calling Sequence: CALL FRRD1B (PP, USETD, GMD, GØD, MULTI, SINGLE, ØMIT, MØDAL, PHIDH,  
PD, PS, PH, SCR1, SCR2, SCR3, SCR4)

PP, USETD, GMD, GØD, PHIDH, PD, PS, PH are GINØ file numbers of their respective data blocks - integer - input.

MULTI -  $MULTI \geq 0$  implies m's are present - integer - input.

SINGLE -  $SINGLE \geq 0$  implies s's are present - integer - input.

ØMIT -  $ØMIT \geq 0$  implies o's are present - integer - input.

MØDAL -  $MØDAL = MØDA$  implies a modal formulation - BCD - input.

SCR1, ..., SCR4 - GINØ file numbers of 4 scratch files - integer - input.

### 4.61.8.3 Subroutine Name: FRRD1C

1. Entry Point: FRRD1C

2. Purpose: To form and decompose "left" hand side of the frequency equation.

3. Calling Sequence: CALL FRRD1C (FRL, FRQSET, MDD, BDD, KDD, I, ULL, LLL, SCR1, SCR2, SCR3,  
SCR4, IGØØD)

FRL, MDD, BDD, KDD, ULL, LLL, SCR1-4 are GINØ file numbers of their respective data blocks - integer - input.

FRQSET - Set id of selected frequency set - integer - output.

I - Current frequency counter - integer - input.

IGØØD -  $IGØØD = 1$  implies a singular matrix - integer - output.

### 4.61.8.4 Subroutine Name: FRRD1D

1. Entry Point: FRRD1D

2. Purpose: To solve for displacements given decomposition factors and loads.

## FUNCTIONAL MODULE FRRD (FREQUENCY RESPONSE - DISPLACEMENT APPROACH)

3. Calling Sequence: CALL FRRD1D (PD,ULL,LLL,SCR1,SCR2,UDVP,I,NLØAD,IGØØD,NFREQ)

PD,ULL,LLL,UDVP,SCR1,SCR2 are GINØ file numbers of their respective data blocks - integer - input.

I - Current frequency count - integer - input.

NLØAD - Number of loads - integer - input.

IGØØD - IGØØD = 1 implies a singular matrix - integer - input.

NFREQ - Total number of frequencies - integer - input.

### 4.61.8.5 Subroutine Name: FRRD1E

1. Entry Point: FRRD1E

2. Purpose: To reorder displacements if necessary.

3. Calling Sequence: CALL FRRD1E (UDVP,UDV,NLØAD,I)

UDVP - GINØ file number of displacements sorted by frequency/load - integer - input.

UDV - GINØ file number of displacements sorted by load/frequency - integer - input.

NLØAD - Number of loads - integer - input.

I - Number of frequencies solved.

### 4.61.8.6 Subroutine Name: FRRD1F

1. Entry Point: FRRD1F

2. Purpose: To solve the uncoupled modal equations.

3. Calling Sequence: CALL FRRD1F (MHH,BHH,KHH,FRL,FRQSET,NLØAD,NFREQ,PH,UHV)

MHH,BHH,KHH,FRL,PH,UHV are GINØ file numbers of their respective data blocks - integer - input.

FRQSET - Selected frequency set id-integer - input.

NFREQ - Number of frequencies in FRQSET - integer - input.

NLØAD - Number of loads (subcases in current execution) - integer - input.

### 4.61.8.7 Subroutine Name: FACTRU

1. Entry Point: FACTRU

2. Purpose: To decompose a matrix by invoking real unsymmetric decomposition  $[A] \Rightarrow [LL][UL]$ .



## FUNCTIONAL MODULE FRRD (FREQUENCY RESPONSE - DISPLACEMENT APPROACH)

3. Calling Sequence: CALL FACTRU (\$n,A,LL,UL,SCR1,SCR2,SCR3)

where A,LL,UL,SCR1,SCR2,SCR3 are the GINØ file numbers of their respective data blocks.

n - statement number to return to if A is singular.

4. Design Requirements:

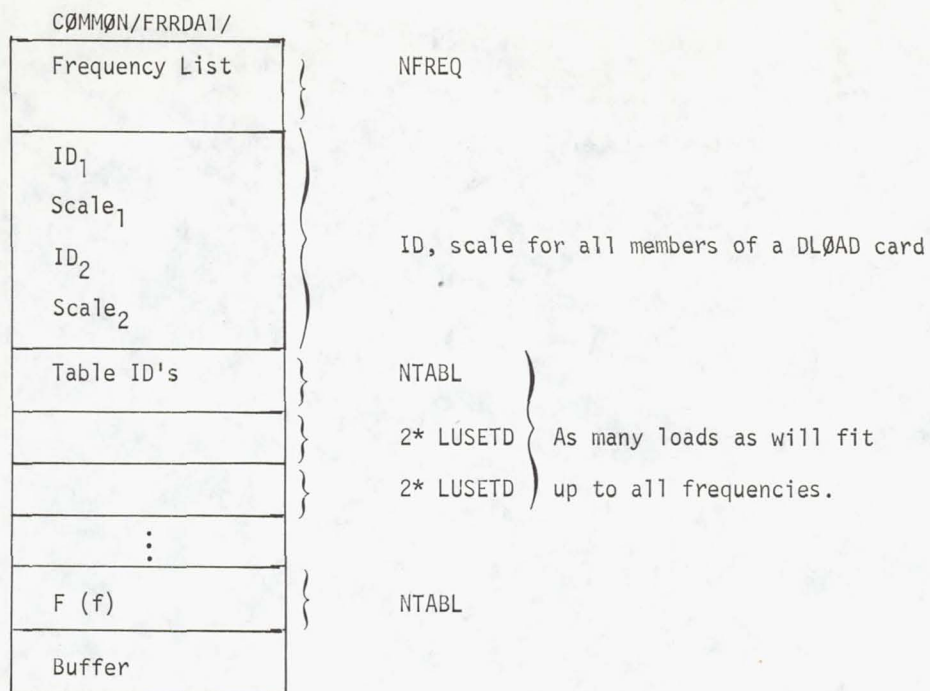
- a) A must have a trailer
- b) A trailer will be written on LL and UL
- c) Open core must be available at /FCTRUX/

### 4.61.9 Design Requirements

Eight scratch files are used by FRRD.

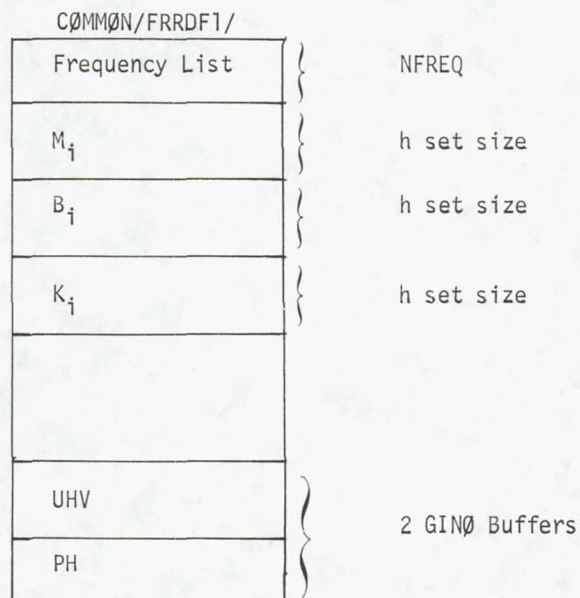
Open core at /FRRDA1/ is used as follows:

## MODULE FUNCTIONAL DESCRIPTIONS



Open core at /FRRDB1/, /FRRDC1/, /FRRDD1/ are used by the matrix routines.

Open core at /FRRDF1/ is used as follows:



### 4.61.10 Diagnostic Messages

Module FRRD may issue the following diagnostic messages:

3005, 3008 and 3045.

# FUNCTIONAL MODULE SDR3 (STRESS DATA RECOVERY - PHASE 3 - SØRT1 TO SØRT2 PROCESSOR)

## 4.62 FUNCTIONAL MODULE SDR3 (STRESS DATA RECOVERY - PHASE 3 - SØRT1 to SØRT2 PROCESSOR)

### 4.62.1 Entry Point: SDR3

### 4.62.2 Purpose

To transpose (perform SØRT2) data blocks containing data prepared for output in the form of ELEMENT-ID-SETS or PØINT-ID-SETS versus TIME-STEP or FREQUENCY-STEP to data prepared for output in the form of TIME-STEP-SETS or FREQUENCY-STEP-SETS versus ELEMENT-ID or PØINT-ID.

#### 4.62.2.1 Example of SØRT1 and SØRT2 Output

Below is a table of ØFP printed output of SDR3 input (SØRT1) and output (SØRT2) data blocks.

#### SDR3 Input Data Block Printed (SØRT1)

| TIME = 1.0 |  | D I S P L A C E M E N T S |      |     |     |     |     |
|------------|--|---------------------------|------|-----|-----|-----|-----|
| POINT-ID   |  | T1                        | T2   | T3  | R1  | R2  | R3  |
| 1          |  | 0.0                       | 4.53 | 0.0 | 0.0 | 0.0 | 0.0 |
| 2          |  | 0.0                       | 5.12 | 0.0 | 0.0 | 0.0 | 0.0 |
| TIME = 2.0 |  | D I S P L A C E M E N T S |      |     |     |     |     |
| POINT-ID   |  | T1                        | T2   | T3  | R1  | R2  | R3  |
| 1          |  | 0.0                       | 4.83 | 0.0 | 0.0 | 0.0 | 0.0 |
| 2          |  | 0.0                       | 5.53 | 0.0 | 0.0 | 0.0 | 0.0 |
| TIME = 3.0 |  | D I S P L A C E M E N T S |      |     |     |     |     |
| POINT-ID   |  | T1                        | T2   | T3  | R1  | R2  | R3  |
| 1          |  | 0.0                       | 6.84 | 0.0 | 0.0 | 0.0 | 0.0 |
| 2          |  | 0.0                       | 7.96 | 0.0 | 0.0 | 0.0 | 0.0 |

#### SDR3 Output Data Block Printed (SØRT2)

| POINT-ID = 1 |     | D I S P L A C E M E N T S |     |     |     |     |
|--------------|-----|---------------------------|-----|-----|-----|-----|
| TIME         | T1  | T2                        | T3  | R1  | R2  | R3  |
| 1.0          | 0.0 | 4.53                      | 0.0 | 0.0 | 0.0 | 0.0 |
| 2.0          | 0.0 | 4.83                      | 0.0 | 0.0 | 0.0 | 0.0 |
| 3.0          | 0.0 | 6.84                      | 0.0 | 0.0 | 0.0 | 0.0 |
| POINT-ID = 2 |     | D I S P L A C E M E N T S |     |     |     |     |
| TIME         | T1  | T2                        | T3  | R1  | R2  | R3  |
| 1.0          | 0.0 | 5.12                      | 0.0 | 0.0 | 0.0 | 0.0 |
| 2.0          | 0.0 | 5.53                      | 0.0 | 0.0 | 0.0 | 0.0 |
| 3.0          | 0.0 | 7.96                      | 0.0 | 0.0 | 0.0 | 0.0 |



## MODULE FUNCTIONAL DESCRIPTIONS

### 4.62.3 DMAP Calling Sequence

SDR3    IN1,IN2,IN3,IN4,IN5,IN6/ØUT1,ØUT2,ØUT3,ØUT4,ØUT5,ØUT6/ \$

### 4.62.4 Input Data Blocks

One to six data blocks in any order desired. Input data blocks to SDR3 which are purged are ignored.

### 4.62.5 Output Data Blocks

One to six data blocks in corresponding order to that of the input data blocks. If SØRT2 is to be performed, there must be an available output data block for the corresponding input data block (Non-Fatal Error if this condition is not met).

### 4.62.6 Parameters

None

### 4.62.7 Method

#### 4.62.7.1 Input and Output Data Block Record Arrangements

Both the input and output data blocks of SDR3 have the following format:

FUNCTIONAL MODULE SDR3 (STRESS DATA RECOVERY - PHASE 3 - SØRT1 TO SØRT2 PROCESSOR)

4.62 FUNCTIONAL MODULE SDR3 (STRESS DATA RECOVERY - PHASE 3 - SØRT1 to SØRT2 PROCESSOR)

4.62.1 Entry Point: SDR3

4.62.2 Purpose

To transpose (perform SØRT2) data blocks containing data prepared for output in the form of ELEMENT-ID-SETS or PØINT-ID-SETS versus TIME-STEP or FREQUENCY-STEP to data prepared for output in the form of TIME-STEP-SETS or FREQUENCY-STEP-SETS versus ELEMENT-ID or PØINT-ID.

4.62.2.1 Example of SØRT1 and SØRT2 Output

Below is a table of ØFP printed output of SDR3 input (SØRT1) and output (SØRT2) data blocks.

SDR3 Input Data Block Printed (SØRT1)

| TIME = 1.0                |     |      |     |     |     |     |
|---------------------------|-----|------|-----|-----|-----|-----|
| D I S P L A C E M E N T S |     |      |     |     |     |     |
| POINT-ID                  | T1  | T2   | T3  | R1  | R2  | R3  |
| 1                         | 0.0 | 4.53 | 0.0 | 0.0 | 0.0 | 0.0 |
| 2                         | 0.0 | 5.12 | 0.0 | 0.0 | 0.0 | 0.0 |
| TIME = 2.0                |     |      |     |     |     |     |
| D I S P L A C E M E N T S |     |      |     |     |     |     |
| POINT-ID                  | T1  | T2   | T3  | R1  | R2  | R3  |
| 1                         | 0.0 | 4.83 | 0.0 | 0.0 | 0.0 | 0.0 |
| 2                         | 0.0 | 5.53 | 0.0 | 0.0 | 0.0 | 0.0 |
| TIME = 3.0                |     |      |     |     |     |     |
| D I S P L A C E M E N T S |     |      |     |     |     |     |
| POINT-ID                  | T1  | T2   | T3  | R1  | R2  | R3  |
| 1                         | 0.0 | 6.84 | 0.0 | 0.0 | 0.0 | 0.0 |
| 2                         | 0.0 | 7.96 | 0.0 | 0.0 | 0.0 | 0.0 |

SDR3 Output Data Block Printed (SØRT2)

| POINT-ID = 1              |     |      |     |     |     |     |
|---------------------------|-----|------|-----|-----|-----|-----|
| D I S P L A C E M E N T S |     |      |     |     |     |     |
| TIME                      | T1  | T2   | T3  | R1  | R2  | R3  |
| 1.0                       | 0.0 | 4.53 | 0.0 | 0.0 | 0.0 | 0.0 |
| 2.0                       | 0.0 | 4.83 | 0.0 | 0.0 | 0.0 | 0.0 |
| 3.0                       | 0.0 | 6.84 | 0.0 | 0.0 | 0.0 | 0.0 |
| POINT-ID = 2              |     |      |     |     |     |     |
| D I S P L A C E M E N T S |     |      |     |     |     |     |
| TIME                      | T1  | T2   | T3  | R1  | R2  | R3  |
| 1.0                       | 0.0 | 5.12 | 0.0 | 0.0 | 0.0 | 0.0 |
| 2.0                       | 0.0 | 5.53 | 0.0 | 0.0 | 0.0 | 0.0 |
| 3.0                       | 0.0 | 7.96 | 0.0 | 0.0 | 0.0 | 0.0 |

## MODULE FUNCTIONAL DESCRIPTIONS

### 4.62.3 DMAP Calling Sequence

SDR3 IN1,IN2,IN3,IN4,IN5,IN6/OUT1,OUT2,OUT3,OUT4,OUT5,OUT6/ \$

### 4.62.4 Input Data Blocks

One to six data blocks in any order desired. Input data blocks to SDR3 which are purged are ignored.

### 4.62.5 Output Data Blocks

One to six data blocks in corresponding order to that of the input data blocks. If SØRT2 is to be performed, there must be an available output data block for the corresponding input data block (Non-Fatal Error if this condition is not met).

### 4.62.6 Parameters

None

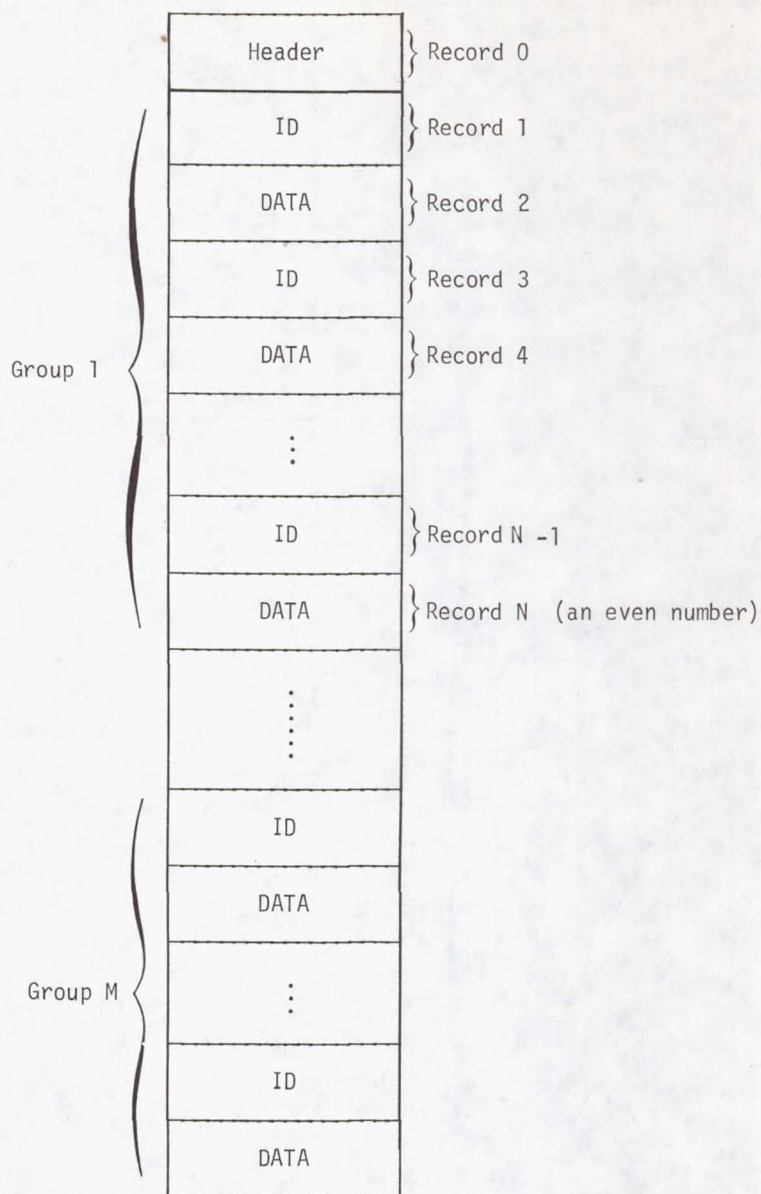
### 4.62.7 Method

#### 4.62.7.1 Input and Output Data Block Record Arrangements

Both the input and output data blocks of SDR3 have the following format:



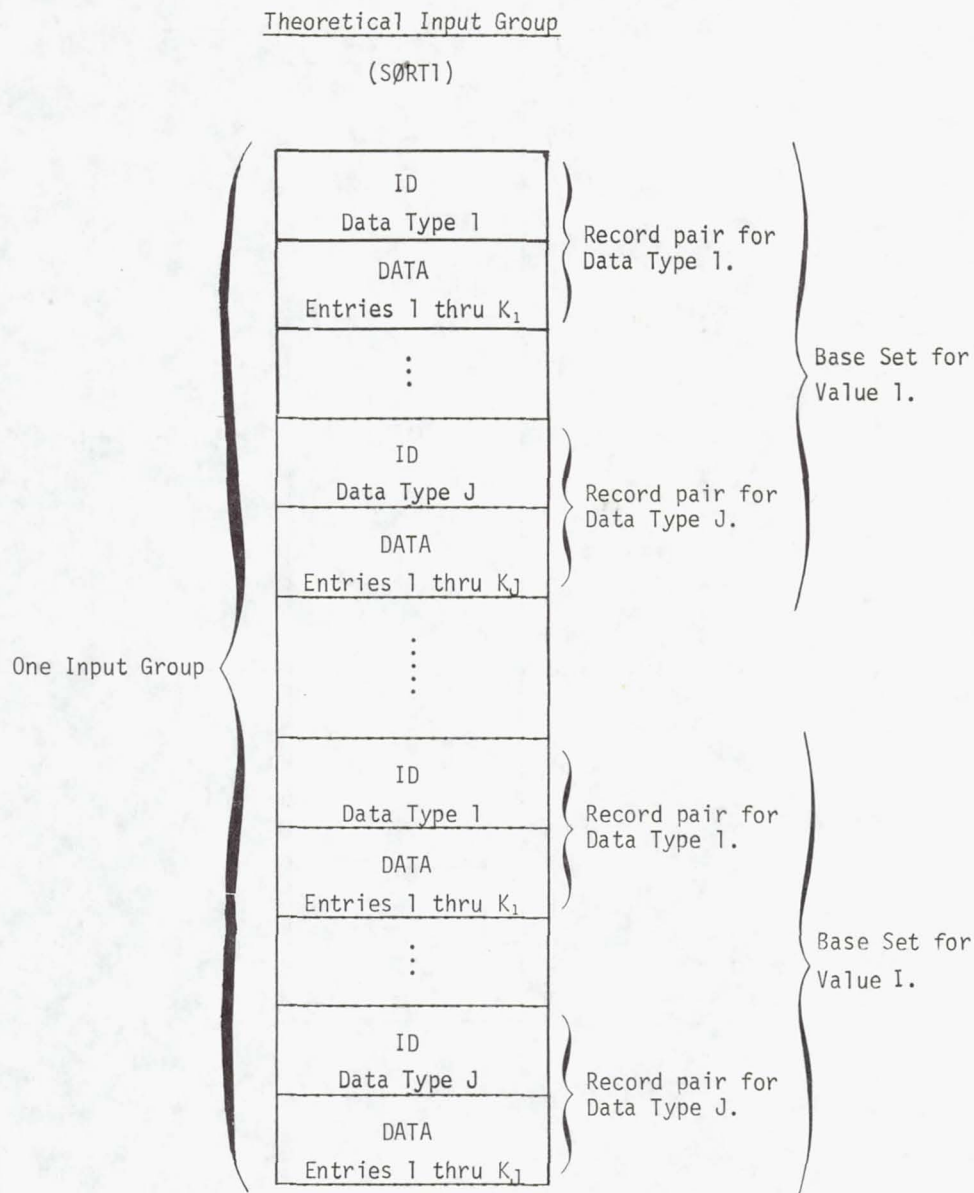
FUNCTIONAL MODULE SDR3 (STRESS DATA RECOVERY - PHASE 3 - SØRT1 TO SØRT2 PROCESSOR)



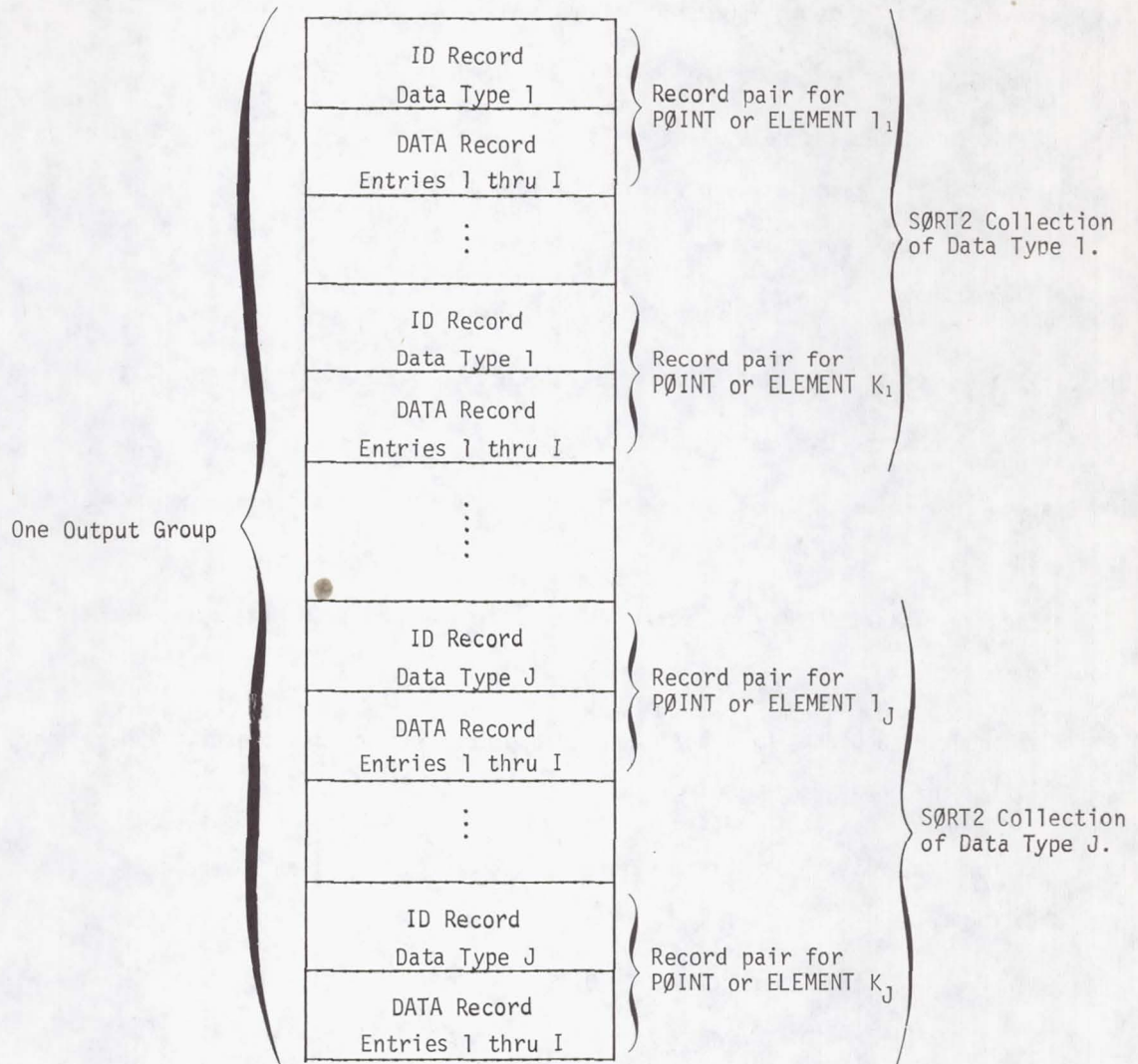
# MODULE FUNCTIONAL DESCRIPTIONS

## 4.62.7.2 Description of a Group

1. An input (SØRT1) data block Group and an output (SØRT2) data block Group are given in the following figures:



Theoretical Output Group  
(SØRT2)





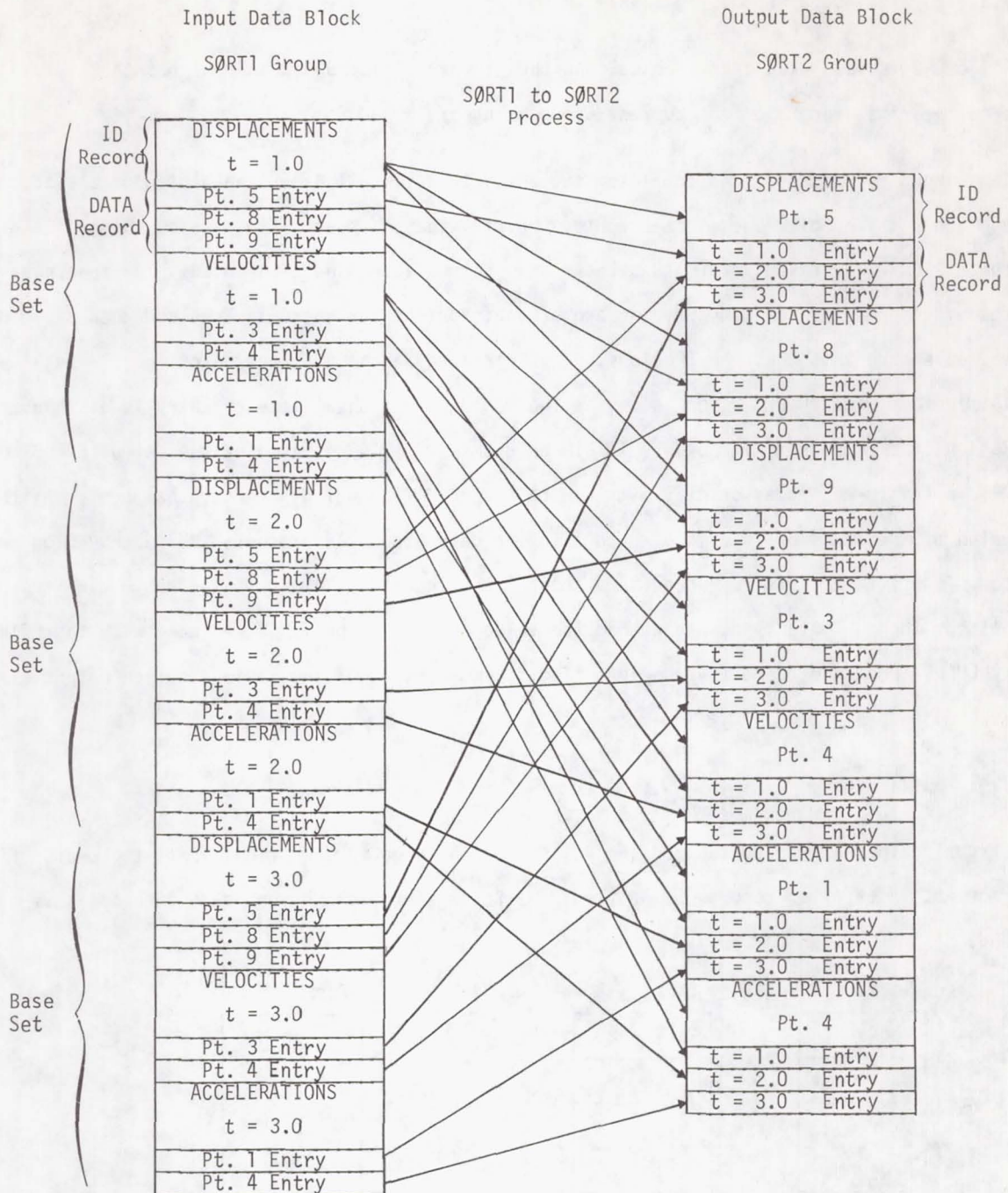
## MODULE FUNCTIONAL DESCRIPTIONS

2. In the above figures each Group is independent of any other Group so far as SDR3 need be concerned.
3. A Group is defined as a collection of successive records belonging to the same subcase.
4. An ID-Record is of a fixed size equal to 146 words.
5. A DATA-Record contains multiple Entries with each Entry being of a length in words specified within the immediately preceding ID-Record.
6.  $I$  = The number of Values (FREQUENCIES or TIMES) present in the Group.
7. A Base Set is a sub-Group of the Group containing data records for one particular Value.
8.  $J$  = The number of different Data Types (DISPLACEMENTS, VELOCITIES, etc.) within a Base Set.
9.  $K_j$  = The number of Entries for Data Type  $j$ .
10. Respective records of any two Base Sets within an input data block Group are of the same size.
11. Respective Entries within respective DATA Records of all Base Sets of an input data block Group begin with the same ELEMENT-ID or POINT-ID.
12. Most input data blocks will contain only one Group having but one Data Type. There is normally more than one Base Set within any Group.
13. A pictorial representation of a SØRT1 to SØRT2 process is given on the next page using the following data:

Values = 3 time steps (1.0, 2.0, 3.0)

Data Types =  $\begin{cases} 1 - \text{Displacements (3 Entries/Value - points 5, 8 and 9)} \\ 2 - \text{Velocities (2 Entries/Value - points 3 and 4)} \\ 3 - \text{Accelerations (2 Entries/Value - points 1 and 4)} \end{cases}$

FUNCTIONAL MODULE SDR3 (STRESS DATA RECOVERY - PHASE 3 - SØRT1 TO SØRT2 PROCESSOR)





## MODULE FUNCTIONAL DESCRIPTIONS

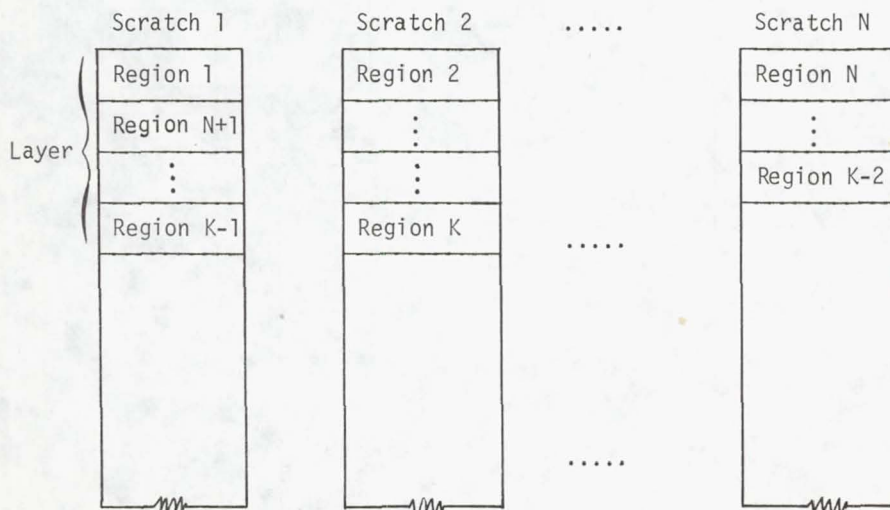
### 4.62.7.3 Physical Data Processing (SØRT1 to SØRT2)

All emphasis is placed on the Group, and thus in performing SØRT2 a Group pointer always points to the first record of the current Group being processed.

Each Group is processed and completed successively until all Groups have been processed. For each Group a loop of J passes is executed. During the  $j^{\text{th}}$  pass of this loop, the  $j^{\text{th}}$  Data Type (note 4.62.7.2) present of the Base Sets will be collected and transposed. The transpose consists of determining how many Entries are present for the current Data Type and then dividing the available core into that many Regions. The Entries of each DATA record for the  $j^{\text{th}}$  Data Type are distributed in Entry order, one each, to the Regions. At the time each Entry is distributed to a Region, the Entry's first word (PØINT-ID or ELEMENT-ID) is replaced by the Value (FREQUENCY or TIME) in the ID-Record associated with the DATA-Record from which the Entry has come. At the conclusion of each pass of this loop, output to the data block can proceed. For each Region an ID-Record is written. This ID-Record is a copy of the input data block ID-Record in the first Base Set for the  $j^{\text{th}}$  Data Type, having had the Value (FREQUENCY or TIME) replaced with the PØINT-ID or ELEMENT-ID of the respective Region. The filled portion of the Region is then output as the DATA-Record.

### 4.62.7.4 Spill Logic

If during the Entry distribution the Regions can hold no more Entries, spill to scratch files is performed. A Layer of records is written, one record for each Region, each time spill is required.





## FUNCTIONAL MODULE SDR3 (STRESS DATA RECOVERY - PHASE 3 - SØRT1 TO SØRT2 PROCESSOR)

At the output stage, if spill to the scratch files has occurred, the Regions in the scratch files are output before the in-core Regions.

### 4.62.8 Subroutines

#### 4.62.8.1 Subroutine Name: SDR3A

1. Entry Point: SDR3A
2. Purpose: To perform all SØRT2 operations when called by the driver routine SDR3.
3. Calling Sequence: CALL SDR3A (ØFPFIL)

ØFPFIL - An array of six words, one for each input data block, each of which is set to zero before the CALL and then reset by SDR3A with a traceback positive integer in the event an error for its respective data block occurred.

### 4.62.9 Design Requirements

1. The design requires that the largest DATA-Record fit in core. If a problem is outputting so many ELEMENT-ID or PØINT-ID Entries for a particular FREQUENCY or TIME that core is insufficient, then more subcases in conjunction with output request sets are recommended.

#### 2. CØMMØN/SDR3ZZ/Z(1)

This common block defines open core for the SDR3 module.

3. SDR3 will open all its scratch files (8).

### 4.62.10 Diagnostic Messages

All errors within SDR3 are considered non-fatal-User Warning type errors. Any error resulting in termination of the SØRT2 process results in the setting of an SDR3 traceback number, an appropriate message, and a call to the ØFP (Output File Processor) which in turn will output the data block in SØRT1 format. If ØFP is unable to output the data block it in turn will call the TABPRT routine, and the data block will be printed.

## FUNCTIONAL MODULE XYTRAN (XY - OUTPUT DATA TRANSLATOR)

### 4.63 FUNCTIONAL MODULE XYTRAN (XY - OUTPUT DATA TRANSLATOR)

#### 4.63.1 Entry Point: XYTRAN

To read the first record of the XYCDB data block (prepared by subroutine IFPIXY of Executive module IFP1); to set xy-output parameters from the serial specifications of this record; to interpret the user curve requests; to locate in the XYTRAN input data blocks (2 thru 6) the data sets containing the requested curve data; to prepare summary and xy-coordinate data for the requested curves and output them to the system output printer and punch units; and to prepare xy-coordinate data and output them to the XYTRAN output data block for direct plotting by the XYPLØT module of those curve requests specified to be plotted.

#### 4.63.3 DMAP Calling Sequences

##### 4.63.3.1 Static Analysis (Rigid Format 1)

1. Stress data recovery output.

XYTRAN XYCDB,ØPG2,ØQG2,ØUGV2,ØES2,ØEF2/XYPLTT/C,N,TRAN/C,N,PSET/V,N,PFILE/V,N,CARDNØ \$

##### 4.63.3.2 Transient Response - Direct Formulation. (Rigid Format 9)

1. Vector data recovery output.

XYTRAN XYCDB,ØUDVC2,,,/XYPLTFA/C,N,FREQ/C,N,DSET/V,N,PFILE/V,N,CARDNØ \$

2. Stress data recovery output.

XYTRAN XYCDB,ØPPC2,ØQPC2,ØUPVC2,ØESC2,ØEFC2/XYPLTF/C,N,FREQ/C,N,PSET/V,N,PFILE/V,N,CARDNØ \$

3. Random response output.

##### 4.63.3.3 Transient Response - Direct Formulation. (Rigid Format 9)

1. Vector data recovery output.

XYTRAN XYCDB,ØUDV2,ØPNL2,,,/XYPLTTA/C,N,TRAN/C,N,DSET/V,N,PFILE/V,N,CARDNØ \$

2. Stress data recovery output.

XYTRAN XYCDB,ØPP2,ØQP2,ØUPV2,ØES2,ØEF2/XYPLTT/C,N,TRAN/C,N,PSET/V,N,PFILE/V,N,CARDNØ \$

##### 4.63.3.4 Frequency Response - Modal Formulation. (Rigid Format 11)

1. Vector data recovery output.

XYTRAN XYCDB,ØUHV2,,,/XYPLTFA/C,N,FREQ/C,N,HSET/V,N,PFILE/V,N,CARDNØ \$



## 2. Stress data recovery output.

XYTRAN XYCDB,ØPPC2,ØQPC2,ØUPVC2,ØESC2,ØEFC2 / XYPLTF / C,N,FREQ / C,N,PSET / V,N,PFILE /  
V,N,CARDNØ \$

## 3. Random Response output

XYTRAN XYCDB,PSDF,AUTØ,,, / XYPLTR / C,N,RAND / C,N,PSET / V,N,PFILE / V,N,CARDNØ \$

## 4.63.3.5 Transient Response - Modal Formulation (Rigid Format 12)

## 1. Vector data recovery output.

XYTRAN XYCDB,ØUHV2,ØPNL2,,, / XYPLTTA / C,N,TRAN / C,N,HSET / V,N,PFILE / V,N,CARDNØ \$

## 4.63.3.6 Aerodynamic - Modal Flutter Analysis (Rigid Format 10)

## 1. VG curve output.

XYTRAN XYCBD,ØVG,,,, / XYPLTCE / C,N,VG / C,N,PSET / V,N,PFILE / V,N,CARDNØ \$

## 4.63.3.7 Heat - Transient Analysis

1. XYTRAN XYCDB,HØPP2,HØQP2,HØUPV2,,HØEF2/HXYPLTT/C,N,TRAN/C,N,PSET/V,N,PFILE/V,N,CARDNØ \$

4.63.4 Input Data Blocks

- XYCDB - XY Output Control Data Block.
- ØUDVC2 - Output displacement vector requests (solution set, SØRT2, complex).
- ØPPC2 - Output load vector requests (solution set, SØRT2, complex).
- ØQPC2 - Output forces of single-point constraint requests (solution set, SØRT2, complex).
- ØUPVC2 - Output displacement vector requests (p set, SØRT2, complex).
- ØESC2 - Output element stress requests (SØRT2, complex).
- ØEFC2 - Output element force requests (SØRT2, complex).
- PSDF - Power Spectral Density Table.
- AUTØ - Autocorrelation function table.
- ØUDV2 - Output displacement vector requests (solution set, SØRT2, real).
- ØPNL2 - Output nonlinear load requests (solution set, SORT2, real).
- HØPP2 }  
ØPP2 } - Output load vector requests (p set, SORT, real).
- HØQP2 }  
ØQP2 } - Output forces of single-point constraint (p set, SØRT2, real).
- ØUPV2 - Output displacement vector requests (p set, SØRT2, real).
- ØES2 - Output element stress requests (SØRT2, real).
- HØEF2 }  
ØEF2 } - Output element force requests (SØRT2, real).
- ØUHV2 - Output displacement vector requests (solution set, SØRT2, complex).



## FUNCTIONAL MODULE XYTRAN (XY - OUTPUT DATA TRANSLATOR)

- ØUHV2 - Output displacement vector requests (solution set, SØRT2, complex).
- ØVG - Output VG curves (SØRT2
- HØUPV2 - Output temperature vector requests (p set, SØRT2, real).
- ØPG2 - Output load vector requests (g set, SØRT2, real).
- ØQG2 - Output forces of single-point constraint requests (g set, SØRT2, real).
- ØUGV2 - Output displacement vector requests (SØRT2, real).

### 4.63.5 Output Data Blocks

- XYPLTFA -
  - XYPLTF -
  - XYPLTR -
  - XYPLTTA -
  - XYPLTT -
  - HXYPLTT -
- } XY-Plot output requests prepared by XYTRAN for direct plotting by XYPLØT.

### 4.63.6 Parameters

- CARDNØ - Input and output-integer-default value = 0. CARDNØ is incremented by one and punched in columns 73-80 of each card punched by XYTRAN.
- PFILE - Input and output-integer-default value = 0. PFILE is incremented by one for each frame XYTRAN defines for output by XYPLØT.
- FREQ - Input-BCD-2-word-constant distinguishes the problem as frequency response.
- TRAN - Input-BCD 2-word-constant distinguishes the problem as transient response.
- RAND - Input-BCD 2-word-constant distinguishes the problem as random response.
- DSET - Input-BCD 2-word-constant distinguishes the input vector as the d set.
- PSET - Input-BCD 2-word-constant distinguishes the input vector as the p set.
- HSET - Input-BCD 2-word-constant distinguishes the input vector as the h set.

# MODULE FUNCTIONAL DESCRIPTIONS

## 4.63.7 Method

4.63.7.1 The following diagram illustrates the process of serially reading through the XYCDB data block's first record and performing the XYTRAN data processing.

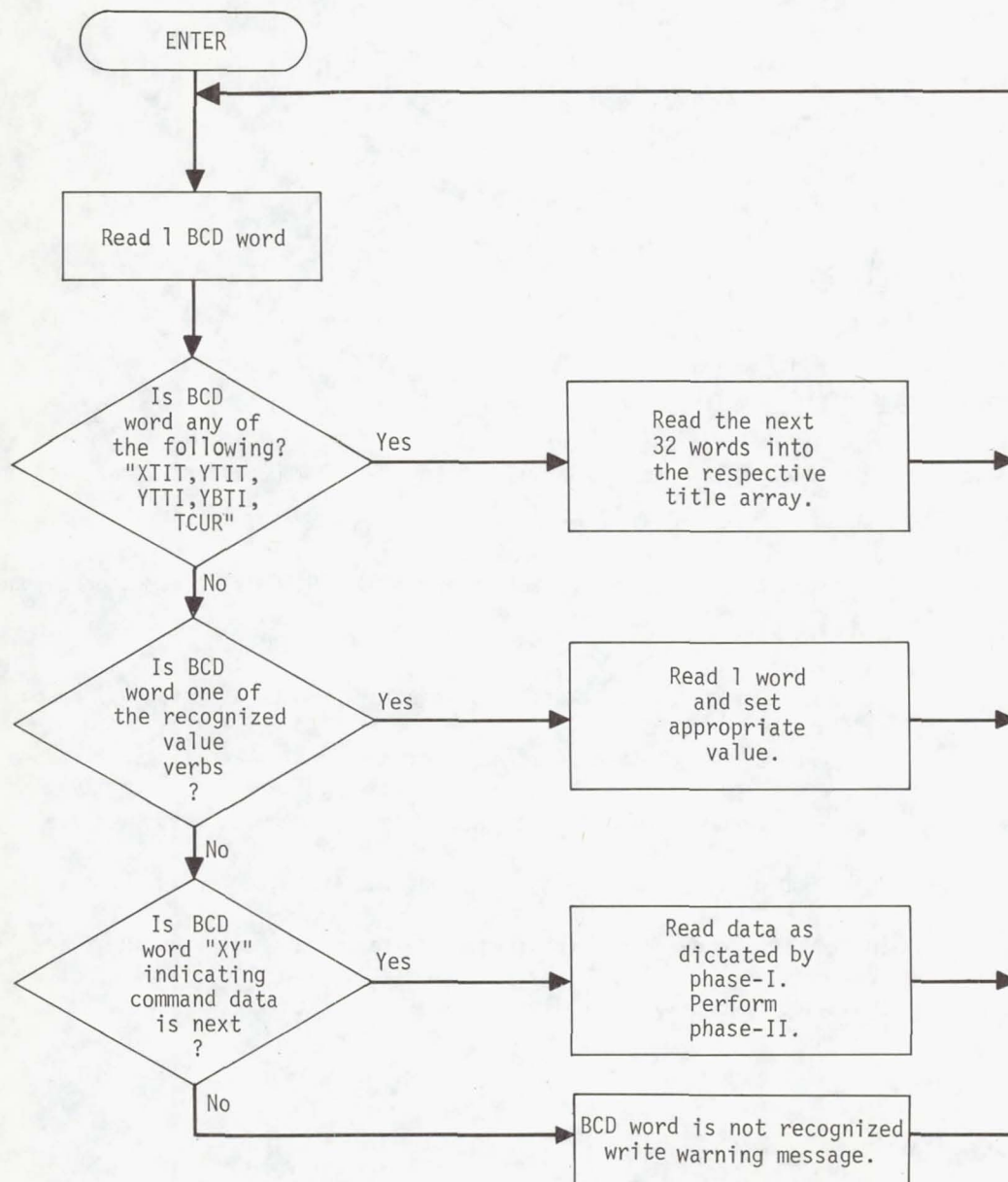


Figure 1. Flowchart for reading the first record of XYCDB

## FUNCTIONAL MODULE XYTRAN (XY - OUTPUT DATA TRANSLATOR)

### 4.63.7.2 Phase I

In Phase I the XYCDB data block is further read to:

1. Determine the type of XY-output curves desired. (Response, Autocorrelation, or Power Spectral Density Function);
2. Determine the type of data (displacements, stresses, etc.,) and subcases desired;
3. Determine which types of XY-output are requested of XYPUNCH, XYPEAK, XYPRINT, XYPAPLOT, and XYPLØT (XY-output requests are described in section 4 of the User's Manual).
4. Determine the point-component curve relationships for a frame.

The data for all curves of a given frame (upper and lower, or whole) are then collected and stored in core.

### 4.63.7.3 Phase II

The operations of Phase II involve the analysis of the curve data in conjunction with the XY-output specifications stored to this point as a set of values, and the computation and setting of dynamic curve limits. When all processing is complete, output to the printer, the punch, and the XYTRAN output data block is accomplished.

### 4.63.8 Subroutines

#### 4.63.8.1 Subroutine Name: XYDUMP

1. Entry Point: XYDUMP
2. Purpose: To perform phase II as described above.
3. Calling Sequence: CALL XYDUMP (IARG, ITYPE).  
  
IARG - 201, GINØ output data block number.  
ITYPE - 1 for RESPONSE, 2 for PSDF, 3 for AUTØ.

#### 4.63.8.2 Subroutine Name: XYFIND

1. Entry Point: XYFIND
2. Purpose: To position one of the XYTRAN input data blocks (2 thru 6) to the beginning of a data set record for a particular ELEMENT-ID or POINT-ID of a specific



## MODULE FUNCTIONAL DESCRIPTIONS

data type.

3. Calling Sequence: CALL XYFIND (\$n<sub>1</sub>, \$n<sub>2</sub>, \$n<sub>3</sub>, MAJID, IDZ)

n<sub>1</sub> = Return taken in the event an end-of-file is sensed when an EOF should not be hit.

n<sub>2</sub> = Return taken in the event an end-of-record is sensed when an end-of-record should not be hit.

n<sub>3</sub> = Return taken if the data requested could not be found.

MAJID = An array of the eleven data type major-IDs.

IDZ = Pointer into the Z array of open core to an ELEMENT-ID or POINT-ID.

### 4.63.8.3 Subroutine Name: XYØUT

1. Entry Point: XYØUT

2. Purpose: To output to the system printer unit an xy-output summary or to output to the system printer and/or punch unit(s) an xy-output coordinate pair.

3. Calling Sequence: CALL XYØUT (IARG, BUFF)

IARG =  $\begin{cases} <0 \text{ implies print summary.} \\ \geq 0 \text{ implies print and/or punch coordinate pair.} \end{cases}$

BUFF = Array containing data to be output.

### 4.63.8.4 Subroutine Name: XYLØG

1. Entry Point: XYLØG

2. Purpose: To analyze the input arguments V1 and V2 and to reset these arguments to powers of ten bracketing the original values. An example follows.

|      |   |           |                     |
|------|---|-----------|---------------------|
| V1   | = | 0.5       | } Input arguments.  |
| V2   | = | 5.6       |                     |
| IARG | = | Undefined |                     |
| V1   | = | 0.1       | } Output arguments. |
| V2   | = | 10.0      |                     |
| IARG | = | 2         |                     |

FUNCTIONAL MODULE XYTRAN (XY - OUTPUT DATA TRANSLATOR)

3. Calling Sequence: CALL XYLØG(V1,V2,IARG)

V1 = Smaller input real variable.

V2 = Large input real variable.

IARG = Number of logarithmic cycles needed to bracket V1 and V2. (Set by XYLØG before return)

4.63.8.5 Subroutine Name: XYTICS

1. Entry Point: XYTICS

2. Purpose: To accept user-specified xy-plot edge-tic specifications and compute actual edge-tic beginning and ending values, their increments to the successive edge-tics, and their scientific values with powers of ten.

3. Calling Sequence: CALL XYTICS (IØUT,ØUT,IARG1,R1,R2,ISKIP,LØG)

IØUT = Integer output array  
ØUT = Real output array } One and the same array.

IARG1 = Number of edge-tic divisions desired by user.

R1 = Minimum coordinate value of edge.

R2 = Maximum coordinate value of edge.

ISKIP = Edge-tic skip count indicating which edge-tics are to have a value printed along with the tic-mark.

LØG = Number of logarithmic cycles. If zero, linear scale is to be calculated.

4.63.8.6 Subroutine Name: XYPRPL

1. Entry Point: XYPRPL

2. Purpose: To process the XYPAPLØT request. The XYTRAN output data block is read and a proper plot is generated for each XYPAPLØT request. Frame numbers are printed as well as titles, and the data are scaled to the size of the page width. Log requests should not be used.

3. Calling Sequence: CALL XYPRPL

4.63.8.7 Subroutine Name: XYCHAR

1. Entry Point: XYCHAR

2. Purpose: To store the points to be plotted into the appropriate line of the output buffer.

## MODULE FUNCTIONAL DESCRIPTIONS

### 3. Calling Sequence: CALL XYCHAR(IRØW,ICØL,CURVCH)

IRØW = Y coordinate of the point to be plotted

ICØL = X coordinate of the point to be plotted

CURVCH = Symbol to be used for the point

### 4.63.8.8 Subroutine Name: XYGRAF

#### 1. Entry Point: XYGRAF

2. Purpose: To print the proper plot for a frame.

#### 3. Calling Sequence: CALL XYGRAF(GRAPH)

GRAPH = Frame border data for the plot

### 4.63.9 Design Requirements

1. The XYTRAN design requires that for a particular frame all of the curve data for the curves of that frame fit in core. If this condition is not possible, one curve at a time will be cancelled, with a warning message output, until the condition is met for the frame in question.

2. The following CØMMØN blocks are used in the subroutines of module XYTRAN.

#### a. CØMMØN/XYWØRK/

This common block contains variables required in the processing of the user output requests.

#### b. CØMMØN/XYTRZZ/

Defines open core for the module

### 4.63.10 Diagnostic Messages

All XYTRAN diagnostic messages are of a USER-WARNING nature. There are no FATAL type error diagnostics. XYTRAN is in all cases expected to make a normal return.



## FUNCTIONAL MODULE RANDØM (RANDOM ANALYSIS MODULE)

### 4.64 FUNCTIONAL MODULE RANDØM (RANDOM ANALYSIS MODULE)

#### 4.64.1 Entry Point: RANDØM

#### 4.64.2 Purpose

To compute power spectral density functions and autocorrelation functions from frequency response data.

#### 4.64.3 DMAP Calling Sequence

RANDØM      XYCDB,DIT,PSDL,ØUPVC2,ØPPC2,ØQPC2,ØESC2,ØEFC2,CASECC/PSDF,AUTØ/V,N,NØRAND    \$

#### 4.64.4 Input Data Blocks

XYCDB      - XY Plotter Control Data Block.  
DIT        - Direct Input Tables.  
PSDL      - Power Spectral Density List.  
ØUPVC2    - Output displacement vector requests (p set, SØRT2, complex).  
ØPPC2     - Output load vector requests (p set, SØRT2, complex).  
ØQPC2     - Output forces of single-point constraint (p set, SØRT2, complex).  
ØESC2     - Output element stress requests (SØRT2, complex).  
ØEFC2     - Output element force requests (SØRT2, complex).  
CASECC    - Case Control Data Table.

Notes: 1. If XYCDB is purged, RANDØM returns.  
2. DIT cannot be purged if PSDL points to tables in DIT.  
3. If PSDL is purged, RANDØM returns.  
4. ØUPVC2, ØPP2, ØQP2, ØESC2, ØEFC2 must contain the requested outputs.  
5. CASECC cannot be purged.

#### 4.64.5 Output Data Blocks

PSDF      - Power Spectral Density Table.  
AUTØ      - Autocorrelation function table.

Notes: PSDF and AUTØ cannot be purged.

## MODULE FUNCTIONAL DESCRIPTIONS

### 4.64.6 Parameters

NØRAND - Output-integer-no default. NØRAND = -1, if no random analysis is requested;  
0, otherwise.

### 4.64.7 Method

#### 4.64.7.1 Overview of the Method

The Random Analysis Module calculates power spectral density functions, autocorrelation functions and mean deviations for selected displacements, loads, forces of single-point constraint, and element forces and stresses.

#### 4.64.7.2 Module Initialization

The following 4 steps of subroutine RAND7 comprise module initialization.

1. The XYCDB must be present or RANDØM returns.
2. A set of RANDPS Bulk Data cards from PSDL must be selected in CASECC or RANDØM returns.
3. The frequency list is extracted from the first non-empty data file.
4. The selected RANDPS cards are read in and stored. The tables referenced are prepared by subroutine PRETAB. The RANDPS (see section 2.4 of the User's Manual) card defines the functions

$$S_{ab}(f) = (x + iy) F_K(f) \quad (1)$$

where a is the subcase id of the excited load set; b is the subcase id of the applied load set ( $a \leq b$ ); (x,y) is a complex number such that if  $a = b$ , then y must be 0.0; and K is the table identification number of a TABRND1 Bulk Data card which defines  $F_K(f)$ , a power spectral density as a tabular function of frequency.

The power spectral density for gust turbulence can also be supplied on a TABRNDG card. Thus

$$S_{ab}(f) = \overline{W_g^2} \frac{2L}{U} \frac{1 + 2(p+1)(kW/U)^2}{[1 + (kW/U)^2]^{p+3/2}}$$

where  $\overline{W_g^2}$ , L, U, p and k are user supplied data and  $W = 2\pi f$ .

If on any RANDPS card  $a \neq b$ , the equations are called coupled, otherwise they are called uncoupled.

# FUNCTIONAL MODULE RANDØM (RANDOM ANALYSIS MODULE)

## 4.64.7.3 The Uncoupled Case

The following eight steps are accomplished in subroutine RAND5.

1. The XYCDB is read for a list of requested points. This list is stored in core. (Subroutine RAND6).
2. Core is allocated for as many points as possible at one word per frequency. If all points will not fit in core, another pass will be made on this file.
3. Compute  $S_{aa}(f)$  at each load change (subroutine TAB).
4. Read in the data from the SØRT2 data block and compute:

$$S_{ja}(f) = |U_j(f)|^2 S_{aa}(f) , \quad (3)$$

where  $U_j(f)$  is the response of the  $j^{th}$  point at frequency  $f$ .

5. These are summed over all loads to form the power spectral density function:

$$S_j(f) = \sum_a S_{ja}(f) , \quad (4)$$

where 'a' runs over all subcase ID's on the RANDPS cards.

6. When all subcases for the points in core have been processed, the mean response  $\bar{q}_j$  is calculated in subroutine RAND3 for each point  $j$ :

$$\bar{q}_j = \left\{ \frac{1}{2} \sum_{i=1}^{N-1} [(S_j(f_i) + S_j(f_{i+1}))](f_{i+1} - f_i) \right\}^{1/2} , \quad (5)$$

where  $N$  = number of frequencies. The mean response is output with both the PSDF and the autocorrelation function.

The zero crossing  $N_0$  is also computed and output with the mean response  $\bar{q}_j$ .  $N_0$  is defined by

$$N_0 = \frac{1}{2\pi} \left[ \int_0^\infty w^2 S_j(w) dw / \int_0^\infty S_j(w) dw \right] . \quad (6)$$

The integral in the denominator is already calculated in this module, and is related to the "mean square response"

$$\overline{q_j^2} = R_j(0) = \int_0^\infty S_j(f) df ; \quad (7)$$

thus, the numerator must be integrated. Compute

$$\overline{r_j^2} = \int_0^\infty f^2 S_j(f) df , \quad \left. \begin{array}{l} \\ \\ \end{array} \right\} \quad (8a)$$

$$\overline{r_j} = \left\{ \frac{1}{2} \sum_{i=1}^{N-1} [S_j(f_i) + S_j(f_{i+1}))](f_{i+1} - f_i) \right\}^{1/2} ,$$



FUNCTIONAL MODULE RANDØM (RANDOM ANALYSIS MODULE)

$$\left. \begin{aligned} \alpha &= (3f_i^2 + 2f_i f_{i+1} + f_{i+1}^2)/6 \\ \beta &= (f_i^2 + 2f_i f_{i+1} + 3f_{i+1}^2)/6 \end{aligned} \right\} \quad (8b)$$

Note that if  $\alpha$  and  $\beta$  are 1.0, the sum for  $\overline{r_j}$  would become the formula for  $\overline{q_j}$ . Then  $N_0 = r_j/q_j$  is the quantity to be output.

7. If PSDF for point  $j$  is requested, one ID and data record are written on the PSDF data block.
8. If an autocorrelation function is requested for point  $j$ , the  $S_j(f)$  are transformed to the time domain to give the autocorrelation function:

$$\begin{aligned} R_j(\tau_m) &= \sum_{i=1}^{N-1} \left\{ \frac{1}{4\pi^2 \tau_m^2} \left[ \frac{S_j(f_{i+1}) - S_j(f_i)}{(f_{i+1} - f_i)} \right] [\cos(2\pi \tau_m f_{i+1}) - \cos(2\pi \tau_m f_i)] + \right. \\ &\quad \left. = \frac{1}{2\pi \tau_m} [S_j(f_{i+1}) \sin(2\pi \tau_m f_{i+1}) - S_j(f_i) \sin(2\pi \tau_m f_i)] \right\} , \end{aligned} \quad (9)$$

FUNCTIONAL MODULE RANDØM (RANDOM ANALYSIS MODULE)

THIS PAGE HAS BEEN LEFT BLANK INTENTIONALLY.

## MODULE FUNCTIONAL DESCRIPTIONS

where  $i$  is the index of the frequencies;  $N$  is highest frequency;  $\tau_m$  is defined by

$$\tau_m = \tau_0 + \frac{m}{M} (\tau_{\max} - \tau_0), \quad (10)$$

where  $\tau_0$  is the starting time lag,  $M$  is the number of time lag intervals, and  $\tau_{\max}$  is the maximum time lag ( $0 < \tau_0 < \tau_m$ ), all of which are defined on a RANDT1 Bulk Data card. Note that if, in Equation 9,  $\tau_m = 0$ , then

$$R_j(\tau_m) = \frac{2}{q_j}. \quad (11)$$

If more points for this data block remain to be done, the file is rewound and another pass is made. If additional file types are requested, steps 1 through 8 outlined above are repeated. This completes the uncoupled case processing.

### 4.64.7.4 The Coupled Case

The following 6 steps are accomplished in subroutine RAND8.

1. A list of unique subcase id's is extracted from the RANDPS cards.
2. The XYCDB is read for a list of requested points. This list is stored in core (subroutine RAND6).
3. An array of core is reserved for each point as follows:  
Let NFREQ = the number of frequencies used and NUNØ be the number of unique subcase id's mentioned on the RANDPS cards. Each point requires  $2 \text{ NFREQ} * \text{NUNØ}$  words of storage.  
As many points as possible are done at once. The data file is read and the data are stored (real/imaginary) for each point until all subcases for all points in core have been processed.
4. For each RANDPS card  $S_{ab}(f_i)$  is looked up for all  $f$  (subroutine TAB).  
For each point in core  $S_j^1(f)$  is computed:

$$S_j^1(f) = H_{ja}(f) S_{ab}(f) \bar{H}_{jb}(f), \quad (12)$$

where  $H_{ja}(f)$  denotes the value of point  $j$  for subcase  $a$ . The bar over the third factor in Equation 12 denotes the complex conjugate. These  $S_j^1(f)$  are summed over all RANDPS cards to form  $S_j(f)$ :



FUNCTIONAL MODULE RANDØM (RANDOM ANALYSIS MODULE)

$$S_j(f) = \left| \sum_{ab} H_{ja}(f) S_{ab}(f) \overline{H}_{jb}(f) \right|. \quad (13)$$

Note that  $S_{ba} = \overline{S_{ab}}$ , the complex conjugate.

5. The mean response and autocorrelation functions are computed as in Equations 9, 10, and 11.

6. If more points for this file remain to be done, the file is rewound and another pass is made.

If additional file types are requested, steps 1 thru 6 are repeated. If not, the coupled case processing is complete.

#### 4.64.8 Subroutines

##### 4.64.8.1 Subroutine Name: RAND7

1. Entry Point: RAND7
2. Purpose: To initialize for both the coupled and uncoupled cases.
3. Calling Sequence: CALL RAND7(IFILE,NFILE,PSDL,DIT,ICØUP,NFREQ,NPSDL,NTAU,LTAB,CASECC,XYCDB).

PSDL,DIT,CASECC,XYCDB are GINØ file numbers for their respective data blocks -

- integer - input.

IFILE - Array of GINØ file numbers of data files to RANDØM - integer - input.

NFILE - Number of files in IFILE - integer - input.

ICØUP - -1 No RANDØM analysis to be done.

- 0 uncoupled algorithm to be used - integer - output.

- 1 coupled algorithm to be used.

NFREQ - Number of frequencies - integer - output.

NPSDL - Number of RANDPS cards selected - integer - output.

NTAU - Number of  $\tau$ 's on RANDT1 cards - integer - output.

LTAB - Amount of core taken up by table storage - integer - output.

CØMMØN/RANDMX/

RAND7 stores most of its output data in /RANDMX/. See core storage layout of /RANDMX/ (section 4.64.9).

## MODULE FUNCTIONAL DESCRIPTIONS

### 4.64.8.2 Subroutine Name: RAND5

1. Entry Point: RAND5
2. Purpose: To compute uncoupled PSDF and AUTØ numbers.
3. Calling Sequence: CALL RAND5(NFREQ,NPSDL,NTAU,XYCDB,LTAB,IFILE,PSDF,AUTØ,NFILE)  
PSDF,AUTØ - GINØ file numbers of respective files - integer -input.  
Other variables are as in RAND7 (Section 4.64.8.1).

### 4.64.8.3 Subroutine Name: RAND8

1. Entry Point: RAND8
2. Purpose: To compute coupled PSDF and AUTØ numbers.
3. Calling Sequence: CALL RAND8 (Same as RAND5).

### 4.64.8.4 Subroutine Name: RAND1

1. Entry Point: RAND1
2. Purpose: To put one ØFP type ID on PSDF and AUTØ.
3. Calling sequence: CALL RAND1 (FILE,MID,TYPE,ID,CØMP,Q).  
FILE - GINØ file number of output file - integer - input.  
MID - File type (PSDF = 4001,AUTØ = 4002) - integer - input.  
TYPE - Curve type - DISP,VELØ,ACCE,LØAD,SPLF,ELFØ, or STRE - BCD, input.  
ID - Point id - integer - input.  
CØMP - Point component - integer - input.  
Q - Mean deviation - real - input.

### 4.64.8.5 Subroutine Name: RAND2

1. Entry Points: RAND2, RAND2A
2. Purpose: To read a SØRT2 type output file until it finds a point id selected by the user in a list.
3. Calling Sequence: CALL RAND2 (FILE,ILIST,LØAD,IF,LEN,LLIST, DATA)  
CALL RAND2A (DATA)  
FILE - GINØ file number of the SØRT2 data file - integer - input.  
ILIST - List of user desired points - input and output.

FUNCTIONAL MODULE RANDØM (RANDOM ANALYSIS MODULE)

LØAD - Subcase id of first data record in ILIST - integer - output.  
IF - Format of data - real/imaginary or magnitude/phase - integer - output.  
LEN - Length of the data line for this record - integer - output.  
LLIST - Length of the ILIST array.  
DATA - Data array - input.

4.64.8.6 Subroutine Name: RAND3

1. Entry Point: RAND3  
2. Purpose: To compute the mean response  $\bar{q}$ .  
3. Calling Sequence: CALL RAND3 (F,S,Q,N)  
F - Array of frequencies - real - input.  
S - Array of power spectral density functions - real - input.  
Q(1) - Mean response - real - output.  
Q(2) - Number of zero crossings - real - output.  
N - Length of the F and S arrays - integer - input.

4.64.8.7 Subroutine Name: RAND4

1. Entry Point: RAND4  
2. Purpose: To compute the autocorrelation function  $R(\tau)$ .  
3. Calling Sequence: CALL RAND4 (F,S,TAU,R,N)  
F,S,N are as described in RAND3.  
TAU -  $\tau$  point at which R is to computed - real - input.  
R - Autocorrelation function at TAU - real - output.

4.64.8.8 Subroutine Name: RAND6

1. Entry Point: RAND6  
2. Purpose: To extract from the XYCDB a list of user requested points for RANDØM output.  
3. Calling Sequence: CALL RAND6 (XYCDB,BUFFER,NPØINT,IZ,INPUT)  
XYCDB - GINØ file number of the XYCDB data block - integer - input.  
BUFFER - GINØ buffer - array - input.  
NPØINT - Number of points requested by the user for this file - integer - output.



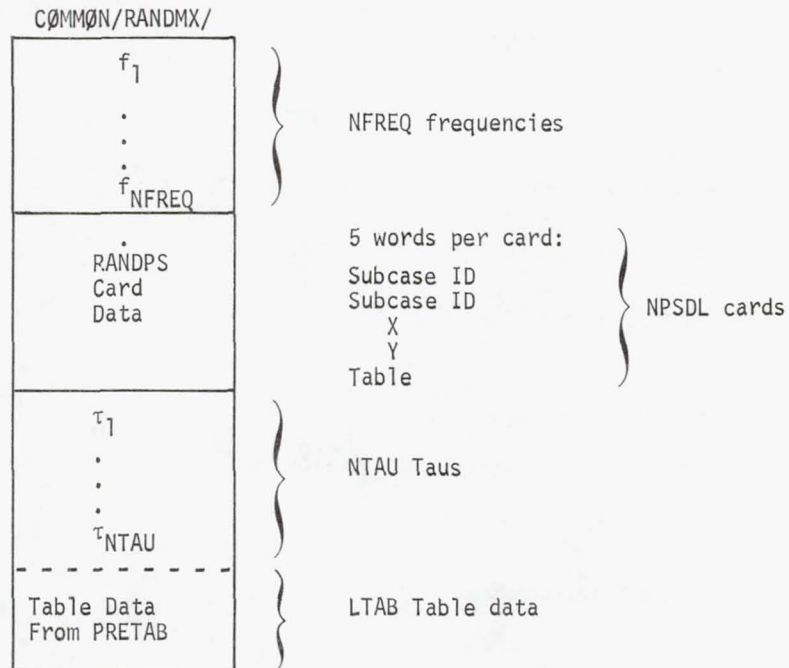
# MODULE FUNCTIONAL DESCRIPTIONS

IZ - Array in which RAND6 stores the list of requests - integer - output.

INPUT - GINØ file number of data file for which list of request is desired.

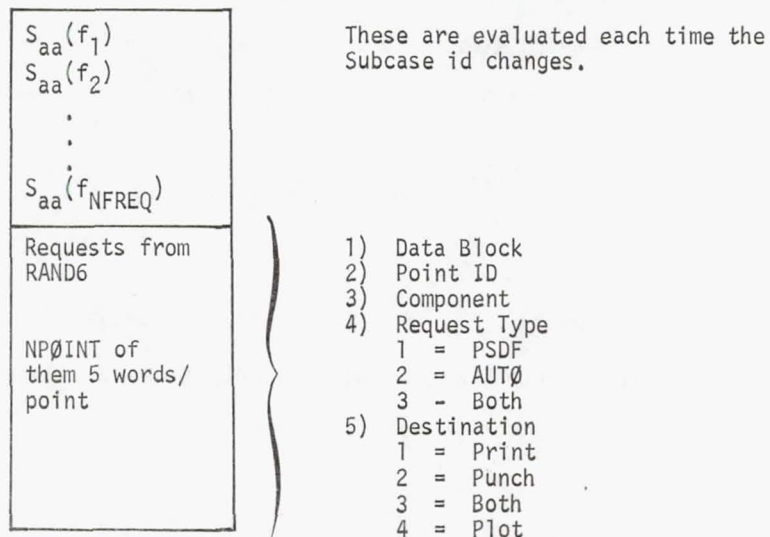
## 4.64.9 Design Requirements

Open Core at /RANDMX/ is arranged as follows:

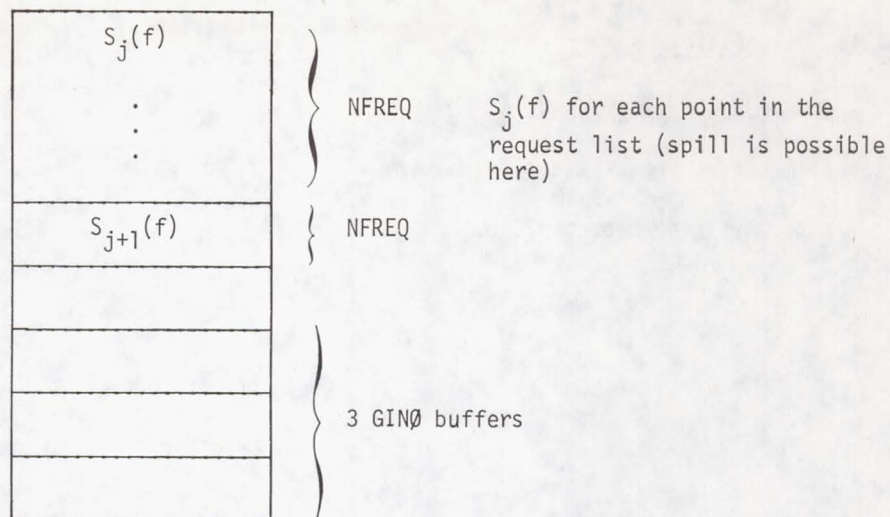


The above data are placed in core by RAND7 and are the same for both the coupled and uncoupled cases. The remaining data are core dependent.

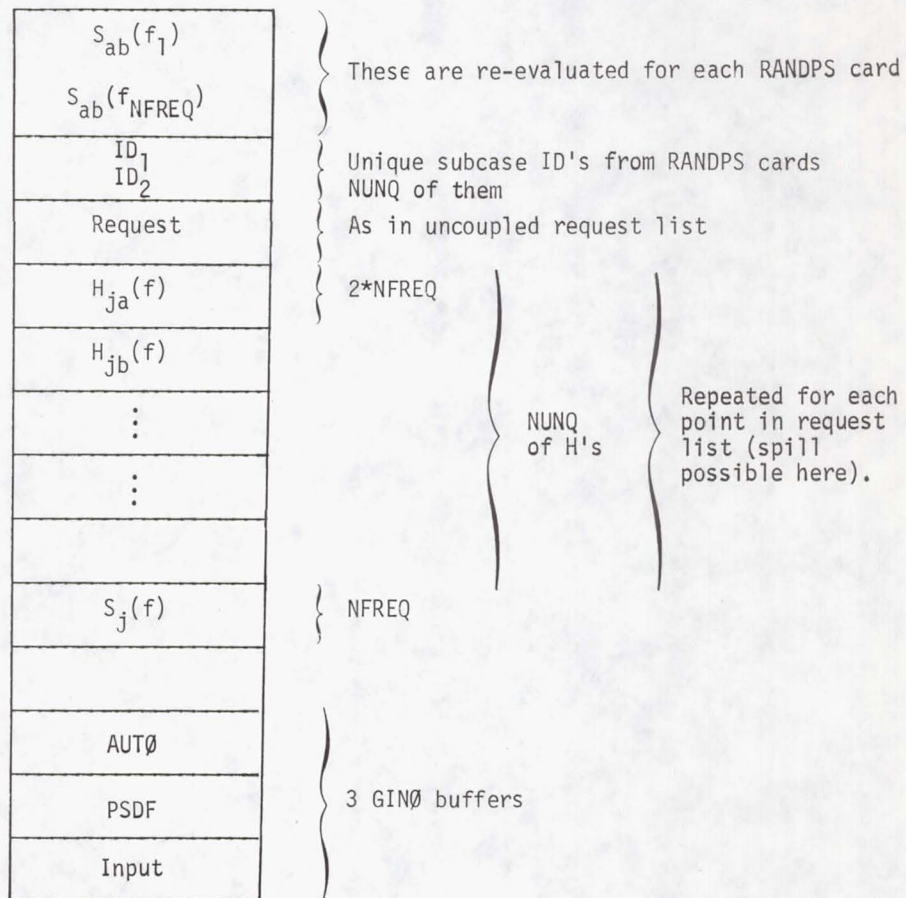
Uncoupled case data:



FUNCTIONAL MODULE RANDØM (RANDOM ANALYSIS MODULE)



Coupled case data:



## MODULE FUNCTIONAL DESCRIPTIONS

### 4.64.10 Diagnostic Messages

RANDØM is defined as an output processor and thus must not stop due to user input error.  
Hence all messages are of a warning nature.

Random may issue message 3048.



## FUNCTIONAL MODULE TRD (TRANSIENT ANALYSIS - DISPLACEMENT)

#### 4.65 FUNCTIONAL MODULE TRD (TRANSIENT ANALYSIS - DISPLACEMENT)

4.65.1 Entry Point: TRD

#### 4.65.2 Purpose

To solve the transient problem.

### 4.65.3 DMAP Calling Sequence

TRD CASECC,TRL,NLFT,DIT,{KDD},{KHH},{BDD},{BHH},{MDD},{MHH},{PD},{PH},{UDVT},{UHVT},{PNLD},{PNLH} / V,N,FORM / V,N,NØUE /

V,N,NØNCUP/V,N,NCØL/C,Y,ISTART \$

#### 4.65.4 Input Data Blocks

CASECC - Case Control Data Table.

TRL - Transient Response List.

NLFT - Non-linear Forcing Table.

DIT - Direct Input Tables.

KHH - Modal stiffness matrix - h set.

KDD - Dynamic stiffness matrix - d set.

BHH - Modal damping matrix - h set.

BDD - Dynamic damping matrix - d set.

MHH - Modal mass matrix - h set.

MDD - Dynamic mass matrix - d set.

PH - Transient Load Matrix.

PD - Linear dynamic load matrix for transient analysis - d set.

Notes:

1. CASECC cannot be purged.
2. TRL cannot be purged.
3. NLFT cannot be purged if nonlinear loads are selected in CASECC.
4. AT least one of the matrices KHH, BHH, or MHH must exist.

## MODULE FUNCTIONAL DESCRIPTIONS

### 4.65.4 Output Data Blocks

- UHVT - Modal transient solution vectors - h set.
- PNLH - Nonlinear loads in modal transient problem - h set.
- UDVT - Displacement, velocity, and acceleration vector matrix in a transient analysis problem - d set.
- PNLD - Non-linear loads in a transient problem - d set.

#### Notes:

1. UHVT cannot be purged.
2. PNLH cannot be purged if nonlinear loads are selected.
3. UHVT will be read if it is not empty and the continue mode will be entered.
4. KHH, MHH, BHH, PH, UHVT, and PNLH fields will contain KDD, MDD, etc., for the direct solution.

### 4.65.6 Parameters

- FØRM - Input-BCD-no default. If FØRM = MØDAL a modal formulation will be used, otherwise a direct formulation will occur.
- NØUE - Input-integer-no default. NØUE indicates the number of extra points used in non-linear load formulation.
- NØNCUP - Input-integer-no default. If NØNCUP = -1 an uncoupled solution will be done.
- NCØL - Input/Output-integer-no default. If NCØL = 0, the initial time for the solution is 0.0. If NCØL > 0, the solution is continued from the specified output time of the previously checkpointed run. (See Section 11.3.2 of the Theoretical Manual for details.)
- ISTART - Input-integer-default = -1. If ISTART < 0, the first starting method is used. If ISTART  $\geq$  0, the second (or alternate) starting method is used. (See Section 11.3.1 of the Theoretical Manual for details.)

### 4.65.7 Method

#### 4.65.7.1 Overview of the Method

The Transient Analysis module integrates, over specified time periods, equations of motion of a structure having time dependent loads. A general structure may be used with real stiffness, mass and damping matrices. Non-linear effects may be calculated by specifying certain loading functions on the free, physical displacements of the system. This analysis is particularly useful when shock loads are applied to a structure. It is also more efficient than frequency analysis or

## FUNCTIONAL MODULE TRD (TRANSIENT ANALYSIS - DISPLACEMENT)

complex eigenvalue analysis when the applied loads are well defined and the frequency characteristics are secondary to damping and peak load characteristics. This analysis is also the only dynamic general system analysis which allows non-linearities.

TRD will also continue the solution from some previous run. This can be used to recover from time-to-go failures or to extend the analysis. (See Section 11.3.2 of the Theoretical Manual for details.).

### 4.65.7.2 Logical Phases of Solution

1. The time increment from the TRL data block is used to identify the times at which the solution is obtained. The initial conditions are assembled.
2. The left hand matrix of the general integration equation, Equation 15 below, and the two right hand matrices are assembled. The triangular decomposition of the left hand matrix is performed.
3. The solution loop of the program may now proceed until the time increment is changed.
  - a. Compute the non-linear load for this time step. Add this load to the load vectors.
  - b. Multiply the displacement vectors into the right hand matrices and add the resultant vectors to the applied load vector.
  - c. Solve for the left hand displacement vector by performing a back substitution into the triangular decomposition of the left hand matrix. If this is an output time step, the velocity and acceleration are computed using differences of the displacement vectors.
  - d. If the time increment changes for the next time step, the program returns to Step 2. If the increment is the same steps 3a thru 3d are repeated.
4. If the equations are in the uncoupled modal formulation form (i.e., no transfer functions, direct input matrices, or non-linear functions), the solution logic is much faster. For each coordinate, the displacement, velocity and acceleration may be computed independently versus time. Steps 2 and 3 are omitted.



## MODULE FUNCTIONAL DESCRIPTIONS

### 4.65.7.3 Algorithms for Each Logical Phase

1. Solution of the coupled equations: The matrix

$$[D] = \left( \frac{1}{\Delta t^2} [M] + \frac{1}{2\Delta t} [B] + \frac{1}{3} [K] \right), \quad (1)$$

is formed and decomposed. The matrix

$$[C] = \left( \frac{2}{\Delta t^2} [M] - \frac{1}{3} [K] \right), \quad (2)$$

is formed and saved. The matrix

$$[E] = \left[ \frac{-1}{\Delta t^2} [M] + \frac{1}{2\Delta t} [B] - \frac{1}{3} [K] \right], \quad (3)$$

is formed and saved.

The solution loops then proceed until a time step change occurs.

The initial conditions presented to the integration are  $\{u_0\}$ ,  $\{\dot{u}_0\}$ ,  $\{u_{-1}\}$ ,  $\{P_0\}$  and  $\{P_{-1}\}$ , where  $\{u_0\}$  and  $\{\dot{u}_0\}$  are the starting displacement and velocity vectors, respectively, specified by the user. Two alternative starting methods have been provided, each having its own advantages. (See Section 11.3.1 of the Theoretical Manual for details.) In both the methods,  $\{u_{-1}\}$  and  $\{P_{-1}\}$  are calculated by the equations

$$\{u_{-1}\} = \{u_0\} - \{\dot{u}_0\} \Delta t, \quad (4)$$

and

$$\{P_{-1}\} = [K]\{u_{-1}\} + [B]\{\dot{u}_0\}. \quad (5)$$

The difference between the two starting methods lies in the different manner in which  $\{P_0\}$  is computed. In the first method (ISTART < 0), the load specified by the user at  $t = 0$  is never used but is replaced by

$$\{P_0\} = [K]\{u_0\} + [B]\{\dot{u}_0\}. \quad (6)$$

# FUNCTIONAL MODULE TRD (TRANSIENT ANALYSIS - DISPLACEMENT)

The starting equation for this method is

$$[D]\{u_1\} = \frac{1}{3}\{P_{-1} + P_0 + P_1\} + \{N_0\} + [C]\{u_0\} + [E]\{u_{-1}\} \quad , \quad (7)$$

where  $\{N_0\}$  is the non-linear load calculated from  $\{u_0\}$ .

In the second (or alternative) starting method ( $ISTART \geq 0$ ), the user input load at  $r = 0$  is included. Thus

$$\{P_0\} = \frac{1}{2} \{P_{ou}\} + [K]\{u_0\} + [B] \{\dot{u}_0\} \quad , \quad (8)$$

where  $\{P_{ou}\}$  is the user-defined load at  $r = 0$ .

The starting equation for this method is

$$[D]\{u_1\} = \frac{1}{6}\{P_{ou} + 2P_1\} + \{N_0\} - \frac{1}{6} [K]\{u_0\} + \frac{1}{2\Delta t} [B]\{u_0\} + \frac{1}{\Delta t^2} [M]\{u_0 + \dot{u}_0 \Delta t\} \quad (9)$$

$\{u_2\}$  through  $\{u_n\}$  are now computed from the general equation:

$$\begin{aligned} [D] \{u_{i+2}\} &= \frac{1}{3} \{P_i + P_{i+1} + P_{i+2}\} + \\ &\{N_{i+1}\} + [C] \{u_{i+1}\} + [E] \{u_i\}. \end{aligned} \quad (10)$$

If non-linear loads are selected, they are evaluated directly at the solution points for time step by the following process. NØLIN1 loads are computed as,

$$P_i(t) = S T (u_j(t)), \quad (11)$$

where  $T$  is a user selected table,  $i$  is the loaded solution point,  $j$  is the deflecting point,  $u_j$  is the previously computed displacement at point  $j$ . NØLIN2 loads are computed as,

$$P_i(t) = S u_j(t) u_k(t), \quad (12)$$

where  $i$ ,  $j$ , and  $k$  are as in NØLIN1 loads.

NØLIN3 loads are computed as,

# MODULE FUNCTIONAL DESCRIPTIONS

$$P_i(t) = \begin{cases} S \{u_j(t)\}A, & u_j(t) > 0 \\ 0 & , u_j(t) \leq 0 \end{cases} \quad (13)$$

NØLIN4 loads are computed as,

$$P_i(t) = \begin{cases} -S \{-u_i(t)\}A, & u_i(t) < 0 \\ 0 & , u_i(t) \geq 0. \end{cases} \quad (14)$$

The user specifies the set of times at which data is to be saved. If the current time is an output time, the displacement vector for time  $t = t_i$  is output.

The velocity vector given by:

$$\{\dot{u}_i\} = \frac{1}{2\Delta t} [\{u_{i+1}\} - \{u_{i-1}\}], \quad (15)$$

is output.

The acceleration vector given by

$$\{\ddot{u}_i\} = \frac{1}{\Delta t^2} [\{u_{i+1}\} + \{u_{i-1}\} - 2\{u_i\}], \quad (16)$$

is output.

If the time step is scheduled to change at  $t_{i+1}$  from  $\Delta t_1$  to  $\Delta t_2$ , the displacement for time  $i+1$  has been calculated.  $\{u_{i-1}\}$ ,  $\{u_i\}$ , and  $\{u_{i+1}\}$  are saved along with  $\{P_{i+1}\}$ . The matrices are formed and decomposed as in Equations 9,10, and 11 for  $\Delta t = \Delta t_2$ .

The following equation is used for computing  $\{u_{i+2}\}$ ,

$$[D] \{u_{i+2}\} = \frac{1}{3} \{P_i^1 + P_{i+1} + P_{i+2}\} + \{N_{i+1}\} + [C] \{u_{i+1}\} + [E] \{u_i^1\}. \quad (17)$$

The vectors  $\{P_i^1\}$  and  $\{u_i^1\}$  in the above equation are calculated as follows. Define

$$\{\dot{u}_{i+1}\} = \frac{1}{\Delta t_1} (\{u_{i+1}\} - \{u_i\}) , \quad (18)$$



FUNCTIONAL MODULE TRD (TRANSIENT ANALYSIS - DISPLACEMENT)

$$\{\dot{u}_{i+1}\} = \frac{1}{\Delta t^2} (\{u_{i+1}\} - 2\{u_i\} + \{u_{i-1}\}) , \quad (19)$$

$$\{\dot{u}_i^1\} = \{\dot{u}_{i+1}\} - \{\dot{u}_{i+1}\} \Delta t_2 , \quad (20)$$

then:

$$\{u_i^1\} = \{u_{i+1}\} - \Delta t_2 \{\dot{u}_{i+1}\} + \frac{\Delta t_2^2}{2} \{\ddot{u}_{i+1}\} , \quad (21)$$

$$\{P_i^1\} = [M] \{\ddot{u}_{i+1}\} + [B] \{\dot{u}_i^1\} + [K] \{u_i^1\} . \quad (22)$$

If the CONTINUE mode is set (NCØL > 0) (see Section 11.3.2 of the Theoretical Manual for details), TRD1C/TRD1C2 will extract the displacement ( $u_n$ ), velocity ( $\dot{u}_n$ ) and acceleration ( $\ddot{u}_n$ ) from the specified time step of the previous run.  $u_1$  (the first displacement of the continued run) is given by

$$[D] \{u_1\} = \frac{1}{3} \{P_{-1} + P_0 + P_1\} + \{N_0\} + [C] \{u_n\} + [E] \{u_{-1}\}, \quad (23)$$

where

$$\{P_0\} = [K] \{u_n\} + [B] \{\dot{u}_n\} + [M] \{\ddot{u}_n\}, \quad (24)$$

$$\{u_{-1}\} = \{u_n\} - \Delta t \{\dot{u}_n\} + \frac{\Delta t^2}{2} \{\ddot{u}_n\}, \quad (25)$$

$$\{\dot{u}_{-1}\} = \{\dot{u}_n\} - \{\ddot{u}_n\} \Delta t , \quad (26)$$

and

$$\{P_{-1}\} = [M] \{\ddot{u}_n\} + [B] \{\dot{u}_{-1}\} + [K] \{u_{-1}\} . \quad (27)$$

2. Solution of Uncoupled Modal Equation: If the method of matrix formulation is modal and no transfer functions or direct input matrices are used, the equations may be solved in a more accurate, more direct manner. The diagonal terms of MHH, BHH, and KHH are stored in core. The following data are necessary to solve the transient behavior of a modal coordinate (1).

# MODULE FUNCTIONAL DESCRIPTIONS

$m_i$  = Modal mass of mode (MHH)

$b_i$  = Modal damping coefficient (BHH)

$K_i$  = Modal stiffness (KHH)

$$\omega_{oi} = (K_i/m_i)^{1/2}, \quad (28)$$

$$\beta_i = \frac{b_i}{2m_i}, \quad (29)$$

$$\omega_i^2 = |\omega_{oi}^2 - \beta^2|, \quad (30)$$

$t_j$  = time of the  $j^{\text{th}}$  time step,

$h_j$  = time increment after the  $j^{\text{th}}$  time,

$f_{ij}$  = applied load on coordinate  $i$  at the  $j^{\text{th}}$  time.

The following coefficients are generated for each distinct time increment and stored in core.

There are four cases. ( $\epsilon = 10^{-5}$  and the subscript  $i$  is implied).

a. If  $\omega_o^2 > \beta^2 + \epsilon$  (underdamped):

$$F = e^{-\beta h} (\cos \omega h + \frac{\beta}{\omega} \sin \omega h), \quad (31)$$

$$G = \frac{1}{\omega} e^{-\beta h} \sin \omega h, \quad (32)$$

$$A = \frac{1}{hk\omega} \{e^{-\beta h} [(\frac{\omega^2 - \beta^2}{\omega_o^2} - \beta h) \sin \omega h - (\frac{2\omega\beta}{\omega_o^2} + h\omega) \cos \omega h] + \frac{2\beta\omega}{\omega_o^2}\}, \quad (33)$$

$$B = \frac{1}{hk\omega} \{e^{-\beta h} [(-\frac{\omega^2 - \beta^2}{\omega_o^2}) \sin \omega h + \frac{2\omega\beta}{\omega_o^2} \cos \omega h] + \omega h - \frac{2\beta\omega}{\omega_o^2}\}, \quad (34)$$

$$F' = -\frac{\omega_o^2}{\omega} e^{-\beta h} \sin \omega h, \quad (35)$$

$$G' = e^{-\beta h} (\cos \omega h - \frac{\beta}{\omega} \sin \omega h), \quad (36)$$

FUNCTIONAL MODULE TRD (TRANSIENT ANALYSIS - DISPLACEMENT)

$$A' = \frac{1}{hk\omega} [e^{-\beta h} \{(\beta + h\omega_0^2) \sin \omega h + \omega \cos \omega h\} - \omega] , \quad (37)$$

$$B' = \frac{1}{hk\omega} [-e^{-\beta h} (\beta \sin \omega h + \omega \cos \omega h) + \omega] , \quad (38)$$

b. If  $|\omega_0^2 - \beta^2| < \epsilon$  (critically damped):

$$F = e^{-\beta h} (1 + \beta h) , \quad (39)$$

$$G = h e^{-\beta h} , \quad (40)$$

$$A = \frac{1}{hk} \left[ \frac{2}{\beta} - \frac{1}{\beta} e^{-\beta h} (2 + 2\beta h + h^2 \beta^2) \right] , \quad (41)$$

$$B = \frac{1}{hk\beta} [-2 + \beta h + e^{-\beta h} (2 + \beta h)] , \quad (42)$$

$$F' = -\beta^2 h e^{-\beta h} , \quad (43)$$

$$G' = e^{-\beta h} (1 - \beta h) , \quad (44)$$

$$A' = \frac{1}{hk} [e^{-\beta h} (1 + h\beta + h^2 \beta^2) - 1] , \quad (45)$$

$$B' = \frac{1}{hk} [1 - e^{-\beta h} (\beta h + 1)] . \quad (46)$$

c. If  $\omega_0^2 < \beta^2 - \epsilon$  (over damped):

$$F = e^{-\beta h} \left( \cosh \omega h + \frac{\beta}{\omega} \sinh \omega h \right) , \quad (47)$$

$$G = \frac{1}{\omega} e^{-\omega h} \sinh \omega h , \quad (48)$$

$$A = \frac{1}{hk\omega} \left\{ e^{-\beta h} \left[ \left( \frac{\omega^2 + \beta^2}{\omega_0^2} - h\beta \right) \sinh \omega h - \left( \frac{2\omega\beta}{\omega_0^2} + h\omega \right) \cosh \omega h \right] + \frac{2\omega\beta}{\omega_0^2} \right\} , \quad (49)$$

$$B = \frac{1}{hk\omega} \left\{ e^{-\beta h} \left[ \frac{\omega^2 + \beta^2}{\omega_0^2} \sinh \omega h + \frac{2\omega\beta}{\omega_0^2} \cosh \omega h \right] + \omega h - \frac{2\beta h}{\omega_0^2} \right\} , \quad (50)$$



# MODULE FUNCTIONAL DESCRIPTIONS

$$F' = -\frac{\omega_0^2}{\omega} e^{-\beta h} \sinh \omega h, \quad (51)$$

$$G' = e^{-\beta h} \left( \cosh \omega h - \frac{\beta}{\omega} \sinh \omega h \right), \quad (52)$$

$$A' = \frac{1}{hk\omega} [e^{-\beta h} \{(\beta + h\omega_0^2) \sinh \omega h + \omega \cosh \omega h\} - \omega], \quad (53)$$

$$B' = \frac{1}{hk\omega} [-e^{-\beta h} (\beta \sinh \omega h + \omega \cosh \omega h) + \omega]. \quad (54)$$

d. If  $|\omega_0| = |\beta| \leq \xi$  (undamped):

$$F = 1, \quad (55)$$

$$G = h, \quad (56)$$

$$A = h^2/3m, \quad (57)$$

$$B = h^2/6m, \quad (58)$$

$$F' = 0, \quad (59)$$

$$G' = 1, \quad (60)$$

$$A' = h/2m, \quad (61)$$

$$B' = h/2m. \quad (62)$$

The equations for each displacement, velocity, and acceleration in terms of the applied loads and previous displacement and velocity are:

$$\xi_{i,j+1} = F_i \xi_{i,j} + G_i \dot{\xi}_{i,j} + A_i f_{i,j} + B_i f_{i,j+1}, \quad (63)$$

$$\dot{\xi}_{i,j+1} = F'_i \xi_{i,j} + G'_i \dot{\xi}_{i,j} + A'_i f_{i,j} + B'_i f_{i,j+1}, \quad (64)$$

$$\ddot{\xi}_{i,j+1} = \frac{P_{i,j+1}}{m_i} + \frac{b_i \dot{\xi}_{i,j+1}}{m_i} - \frac{K_i \xi_{i,j+1}}{m_i}. \quad (65)$$

## FUNCTIONAL MODULE TRD (TRANSIENT ANALYSIS - DISPLACEMENT)

### 4.65.8 Subroutines

Utility routines PRETAB, TAB, SSG2A, CALCV, SSG2B, ADD, SDCOMP, and DECOMP are used. See subroutine descriptions, Section 3 for details.

4.65.8.1 Subroutine Name: TRD1A Single Precision  
TRD1A2 Double Precision

1. Entry Point: TRD1A, TRD1A2
2. Purpose: To assemble the loads at all time steps.
3. Calling Sequence: CALL TRD1A (CASECC, TRL, IC, NLFTP, NGRDUP, MDAL)  
CALL TRD1A2 (CASECC, TRL, IC, NLFTP, NGRDUP, MDAL)

TRL, CASECC - GINØ file numbers of their respective data blocks - integer - input.  
IC - GINØ file number of initial condition matrix - integer - input.  
NLFTP - Non-linear load set id selected in CASECC - integer - input - output.  
NGRDUP - Number of time step changes - integer - output.  
MDAL - If MDAL = 1, a modal formulation is being used - integer - input.

4.65.8.2 Subroutine Name: INITL Single Precision  
INITL2 Double Precision

1. Entry Point: INITL, INITL2
2. Purpose: To form [C] and [E] matrices and to form and decompose the [D] matrix.
3. Calling Sequence: CALL INITL (OFFSET, DELTA)  
CALL INITL2 (OFFSET, DELTA)

COMMON/TRDXX/ See Section 4.65.8.3.

OFFSET - Length of reserved area of core - integer - input.  
DELTA - Current time increment - real - input.

4. Method: INITL/INITL2 will choose between symmetric and unsymmetric decomposition based on the trailers of the input matrices [K], [B], and [M]. It will also set TSYM in /TRDXX/ to inform the remaining routines.



## MODULE FUNCTIONAL DESCRIPTIONS

4.65.8.3 Subroutine Name: TRD1C Single Precision  
TRD1C2 Double Precision

1. Entry Point: TRD1C, TRD1C2
2. Purpose: To solve the coupled equations.
3. Calling Sequence: CALL TRD1C (IC,PAPPLD,NGRØUP,NLFTP,UDV,I,SCR1,DIT,NLFT,NØUE,MØDA1,PNL)  
CALL TRD1C2 (IC,PAPPLD,NGRØUP,NLFTP,UDV,I,SCR1,DIT,NLFT,NØUE,MØDA1,PNL)

IC,NGRØUP } - Are as described in TRD1A - integer - input.  
NLFTP

UDV,DIT } - Are GINØ file numbers of their respective data blocks - integer - input.  
NLFT,PNL

SCR1 - GINØ file number of a scratch file.

PAPPLD - GINØ file number of applied loads - integer - input.

NØUE - Module parameter.

MØDA1 - -1 if FØRM ≠ MØDAL. 1 if FØRM = MØDAL - integer - input.

I - Current loop count. Runs from 1 to number of time step changes - integer - input.

CØMMØN/TRDXX/IK(7),IM(7),IB(7),C,LLL,ULL,E,SCR1,SCR2,IØPEN,ISYM,TØ,NØPD,ISPNL

IK(7) - Matrix control block for K matrix.

IM(7) - Matrix control block for M matrix.

IB(7) - Matrix control block for B matrix.

C - GINØ file number for C matrix.

LLL,ULL - GINØ file numbers for decomposition products of D matrix.

E - GINØ file number for E matrix.

SCR1,SCR2 - GINØ file numbers for 2 scratch files.

IØPEN - 1 implies C,ULL,LLL, and E are open.  
0 implies C,ULL,LLL, and E are closed.

ISYM - 1 implies unsymmetric decomposition used.  
0 implies symmetric decomposition used.

TØ - Initial time (usually 0.0).

NØPD - True if PD does not exist.  
False if PD exists.

ISPNL - 0 if PNLD is not to be formed.  
1 if PNLD is to be formed.

4.65.8.4 Subroutine Name: FØRM1 Single Precision  
FØRM12 Double Precision

1. Entry Point: FØRM1, FØRM12
2. Purpose: To compute  $\{u_{-1}\}$ ,  $\{P_0^1\}$ , and  $\{P_{-1}\}$  for starting the integration procedure.



# FUNCTIONAL MODULE TRD (TRANSIENT ANALYSIS - DISPLACEMENT)

3. Calling Sequence: CALL FØRM1 (UØ,UDØTØ,UI,PØ,PI,DELTAT,IBUF)  
CALL FØRM12 (UØ,UDØTØ,UI,PØ,PI,DELTAT,IBUF)

UØ - Array of core containing  $\{u_0\}$  - real - input.  
UDØTØ - Array of core containing  $\{u_0\}$  - real - input.  
UI - Array of core for storage of  $\{u_{-1}\}$  - real - output.  
PØ - Array of core for storage of  $\{P_0^1\}$  - real - output.  
PI - Array of core for storage of  $\{P_{-1}\}$  - real - output.  
DELTAT - Current time step size - real - input.  
IBUF - GINØ buffer.

4.65.8.5 Subroutine Name: MATVEC Single Precision  
MATVC2 Double Precision

1. Entry Point: MATVEC, MATVC2

2. Purpose: To form the product  $\{X\} = \{X\} + [A] \{Y\}$  where  $[A]$  is a matrix and  $\{Y\}$  is a vector.

3. Calling Sequence: CALL MATVEC (Y,X,FILEA,IBUF)  
CALL MATVC2 (Y,X,FILEA,IBUF)

Y - Array of core containing Y array real - input.  
X - Array of core containing X array real - input/output.  
FILEA - Matrix control block for A. If  $FILEA(1) \leq 0$ , MATVEC will return.  
IBUF - GINØ buffer. If  $IBUF \leq 0$ , MATVEC/MATVC2 will assume the file is already in core.  
COMMON/TRDXX/ (see Section 4.65.8.3).

4.65.8.6 Subroutine Name: STEP Single Precision  
STEP2 Double Precision

1. Entry Point: STEP, STEP2

2. Purpose: To integrate forward 1 time step.

3. Calling Sequence: CALL STEP (U2,U1,UØ,P,IBUF)  
CALL STEP2 (U2,U1,UØ,P,IBUF)

U2 - Array which will contain  $\{u_{i+2}\}$  - real - output.  
U1 - Array containing  $\{u_{i+1}\}$  - real - input.  
UØ - Array containing  $\{u_i\}$  - real - input.  
P - Array containing combined load - real - input.  
IBUF - GINØ buffer - input.  
COMMON/TRDXX/ (see Section 4.65.8.3).

## MODULE FUNCTIONAL DESCRIPTIONS

### 4.65.8.7 Subroutine Name: INTFBS

1. Entry Point: INTFBS
2. Purpose: To perform the forward-backward substitution necessary to solve the system of equations:  $[A] \{Y\} = \{X\}$  for  $\{Y\}$  if  $[A]$  was unsymmetric.

3. Calling Sequence: CALL INTFBS (X,Y,IBUF)

X - Load vector (i.e., right hand side) - real - input.

Y - Solution vector - real - output.

IBUF - GINØ buffer.

COMMON/TRDXX/ (see Section 4.64.8.3).

COMMON/INFBS/FILEL(7),FILEU(7)

FILEL - Matrix control block of the lower triangular factor from the decomposition of A.

FILEU - Matrix control block of the upper triangular factor from the decomposition of B.

### 4.65.8.8 Subroutine Name: TRD1D Single Precision TRD1D2 Double Precision

1. Entry Point: TRD1D, TRD1D2
2. Purpose: To compute the non-linear loads at each time step.

3. Calling Sequence: CALL TRD1D  
CALL TRD1D2

COMMON/TRDD1/NLFT,DIT,NLFTP,NØUT,ICØUNT,ILØØP,MØDA1,NZ,ICØRE,IU2,IP4,IPNL(7),NMØDES,  
NSTEP,PNL

The variables DIT,NLFT,NLFTP,MØDA1 and PNL are defined as in TRD1C (see section 4.65.8.4).

NØUT - Output interval - integer - input.

ICØUNT - Current time step counter - integer - input.

ILØØP - Current time change counter - integer - input.

NZ - Length of open core - integer - input.

ICØRE - Pointer to first unused cell of open core - integer - input.

IU2 - Pointer to displacement vector - 1 - integer - input.

IP4 - Pointer to load area -1 - integer - input.

IPNL - Matrix control block for PNL - integer - input/output.

NMØDES - Number of modes if modal formulation is being used - integer - input.

NSTEP - Number of times steps for this time increment - integer - input.



## FUNCTIONAL MODULE TRD (TRANSIENT ANALYSIS - DISPLACEMENT)

### 4.65.8.9 Subroutine Name: TRD1E

1. Entry Point: TRD1E
2. Purpose: To solve the uncoupled modal equations.
3. Calling Sequence: CALL TRD1E (MHH,BHH,KHH,PH,UHV,NGRØUP)  
MHH,BHH,KHH, - GINØ file numbers of their respective data blocks - integer- input.  
PH,UHV  
NGRØUP - Number of time step changes - integer - input.

### 4.65.8.10 Subroutine Name: FØRM2 Single Precision FØRM22 Double Precision

1. Entry: FØRM2, FØRM22
2. Purpose: To compute  $\{u_i^{\cdot}\}$  and  $\{P_i^{\cdot}\}$  when changing time steps. (See Equations 15 through 25).
3. Calling Sequence: CALL FØRM2 (UDDIP1,UDIPI,UIP,PIP,IBUF)  
CALL FØRM22 (UDDIP1,UDIPI,UIP,PIP,IBUF)  
UDDIP1 - Array of core containing  $\{\ddot{u}_{i+1}\}$  - real - input.  
UDIPI - Array of core containing  $\{\dot{u}_{i+1}\}$  - real - input.  
UIP - Array of core containing  $\{u_i^{\cdot}\}$  - real - output.  
PIP - Array of core containing  $\{P_i^{\cdot}\}$  - real - output.  
IBUF - GINØ buffer  
CØMMØN/TRDXX/ (see Section 4.65.8.3)

### 4.65.8.11 Subroutine Name: FBSINT

1. Entry Point: FBSINT
2. Purpose: To perform the same functions as INTFBS (Section 4.65.8.7) if [A] is symmetric. Subroutine FBS21 is called to process data in mixed precisions.
3. Calling Sequence: Identical to INTFBS.



## MODULE FUNCTIONAL DESCRIPTIONS

### 4.65.9 Design Requirements

1. Open core at /TRDIX/ is illustrated as follows:

|  |   |
|--|---|
| CØMMØN/TRDIX/                                |   |
| Open core for<br>DECØMP or SDCØMP<br>and ADD |   |
| Number of Steps                              | } |
| $\Delta t$                                   |   |
| Output Interval                              |   |

Repeated for each time step change

FUNCTIONAL MODULE TRD (TRANSIENT ANALYSIS - DISPLACEMENT)

4.65.8.10 Subroutine Name: TRD1E

1. Entry Point: TRD1E
2. Purpose: To solve the uncoupled modal equations.
3. Calling Sequence: CALL TRD1E (MHH,BHH,KHH,PH,UHV,NGRØUP)

MHH,BHH,KHH,  
PH,UHV - GINØ file numbers of their respective data blocks - integer - input.

NGRØUP - Number of time step changes - integer - input.

4.65.8.11 Subroutine Name: FØRM2

1. Entry Point: FØRM2
2. Purpose: To compute  $\{u'_i\}$  and  $\{P'_i\}$  when changing time steps. (See Equations 23 through 28)
3. Calling Sequence: CALL FØRM2 (UDDIP1,UDIP1,UIP,PIP,IBUF)

UDDIP1 - Array of core containing  $\{\ddot{u}_{i+1}\}$  - real - input.

UDIP1 - Array of core containing  $\{\dot{u}_{i+1}\}$  - real - input.

UIP - Array of core containing  $\{u'_i\}$  - real - output.

PIP - Array of core containing  $\{P'_i\}$  - real - output.

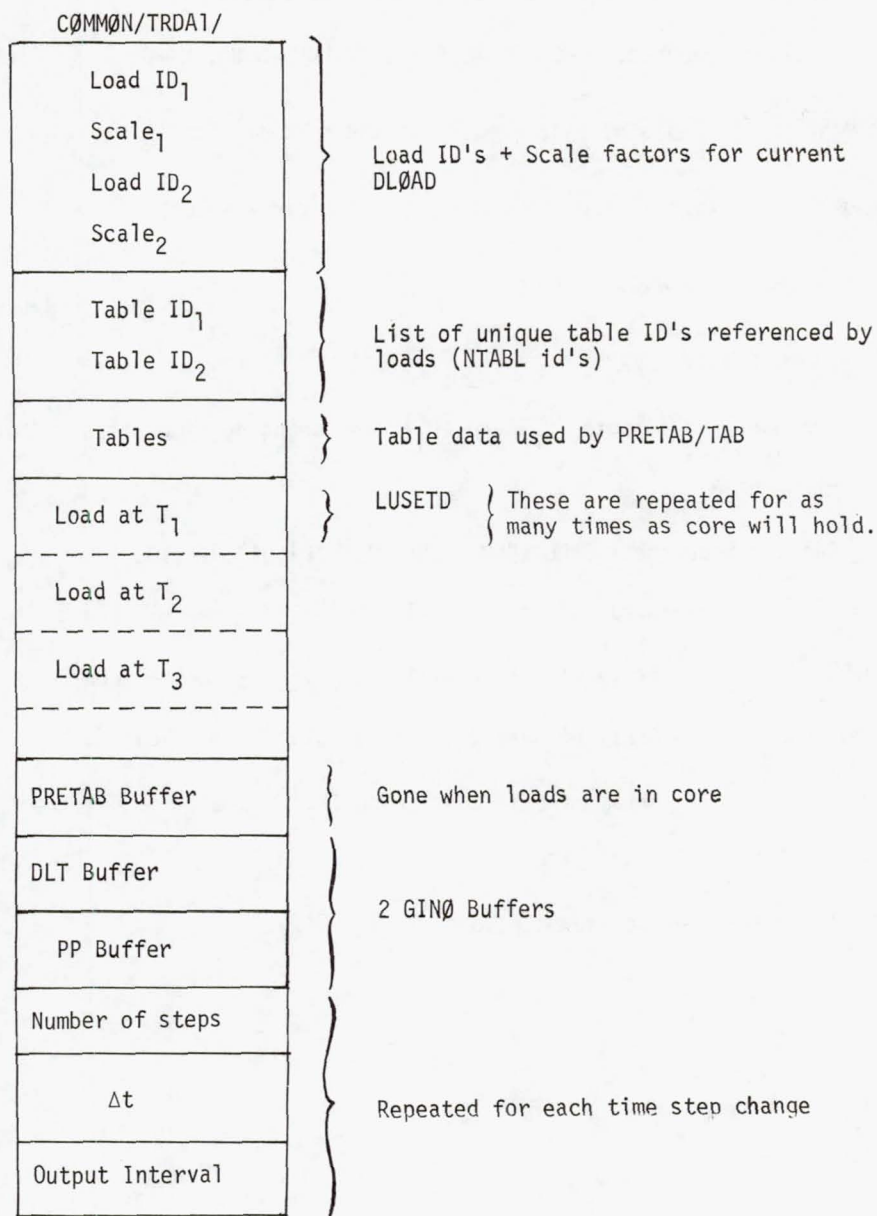
IBUF - GINØ buffer

CØMMØN/TRDXX/ (See section 4.65.8.4)

# MODULE FUNCTIONAL DESCRIPTIONS

## 4.65.9 Design Requirements

1. Open core at /TRDA1/ is illustrated as follows:

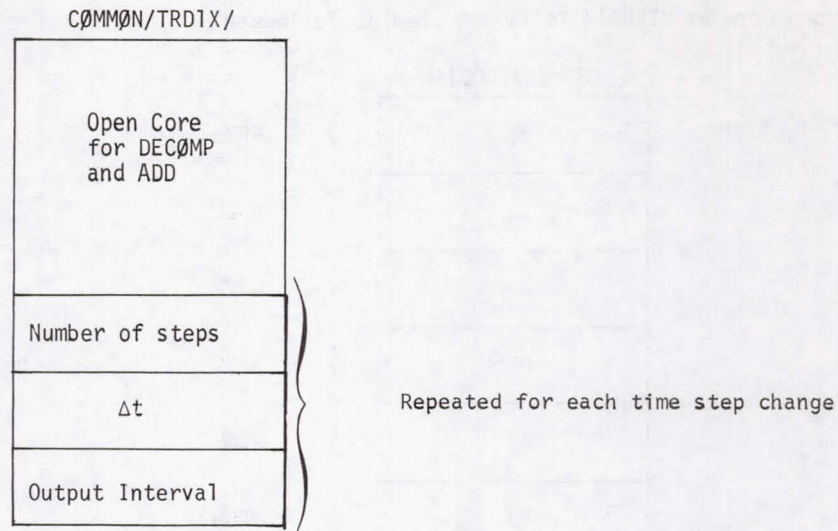


There must be enough core to assemble one time step load.



FUNCTIONAL MODULE TRD (TRANSIENT ANALYSIS - DISPLACEMENT)

2. Open Core at /TRDIX/ is illustrated as follows:



# MODULE FUNCTIONAL DESCRIPTIONS

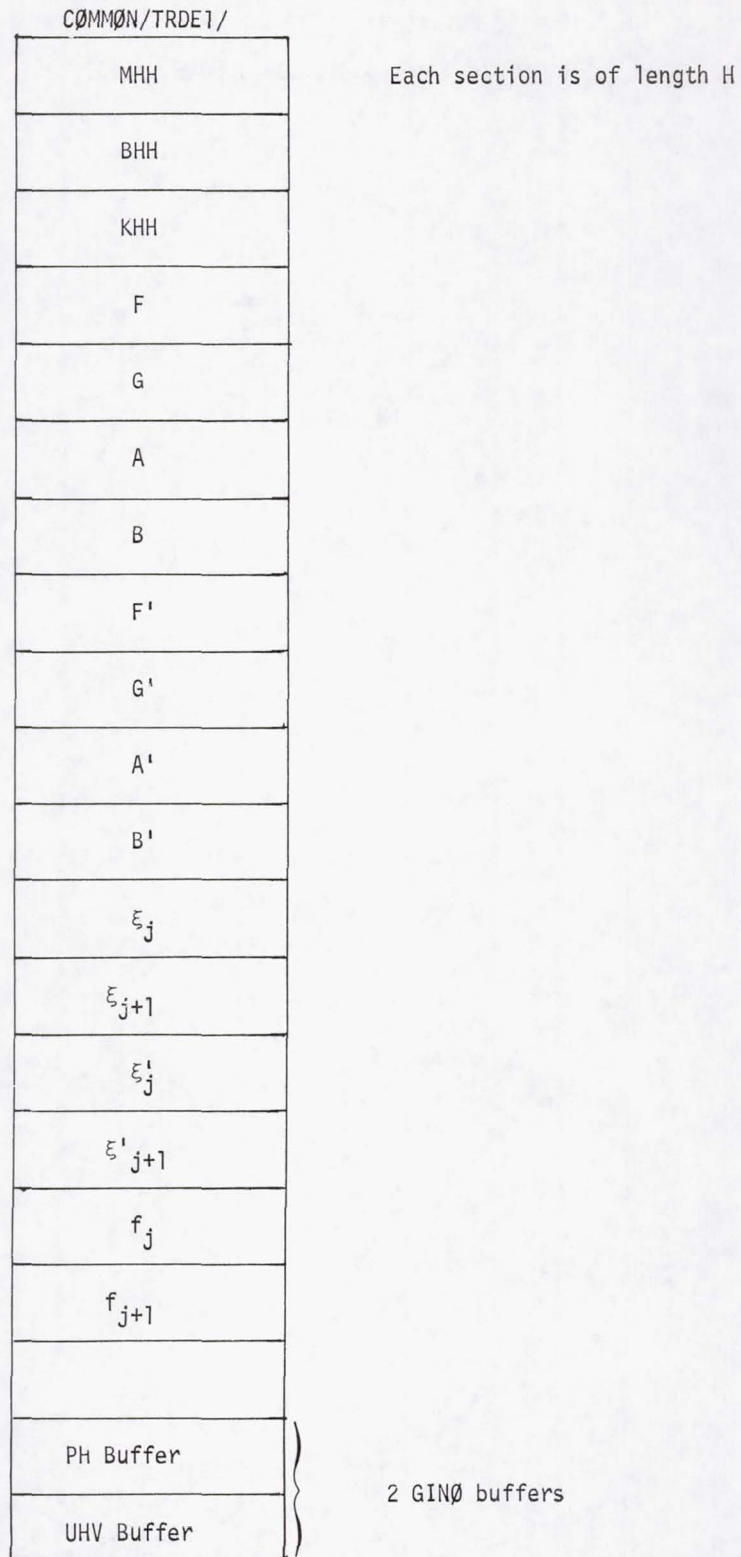
This table is at the bottom of open core through the module.

3. Open Core at /TRDC1/ is illustrated as follows:

| COMMON/TRDC1/   |   |   |
|-----------------|---|---|
| u <sub>1</sub>  | } | NRØW  |
| u <sub>2</sub>  | } | NRØW  |
| u <sub>3</sub>  | } | NRØW  |
| P1              | } | NRØW  |
| P2              | } | NRØW  |
| P3              | } | NRØW  |
| P4              | } | NRØW  |
| Type            | } | 5 words for each non-linear load card selected. |
| Table ID's      | } | Table ID's selected on NØLIN cards.             |
| Table's for TAB |   |   |
| Tab Buffer      | } | Used only if non-linear loads are selected.     |
| C Buffer        | } | GINØ Buffers                                    |
| D Buffer        |   |   |
| ULL Buffer      |   |   |
| LLL Buffer      |   |   |
| Solution Buffer |   |   |
| Load Buffer     |   |   |
| Utility Buffer  |   |   |

FUNCTIONAL MODULE TRD (TRANSIENT ANALYSIS - DISPLACEMENT)

4. Open Core at /TRDE1/ is illustrated as follows:





## MODULE FUNCTIONAL DESCRIPTIONS

### 4.65.10 Diagnostic Messages

TRD may issue the following messages

3001, 3002, 3003, 3005, 3007, 3008, 3031, 3044, 3045, 3046.

## FUNCTIONAL MODULE GKAM (GENERAL K ASSEMBLER MODAL)

### 4.66 FUNCTIONAL MODULE GKAM (GENERAL K ASSEMBLER MODAL)

#### 4.66.1 Entry Point: GKAM

#### 4.66.2 Purpose

To assemble the modal mass, damping and stiffness matrices.

#### 4.66.3 DMAP Calling Sequence

```
GKAM  USETD,PHIA,MI,LAMA,DIT,M2DD,B2DD,K2DD,CASECC / MHH,BHH,KHH,PHIDH / V,N,NØUE /  
      C,Y,LMØDES / C,Y,LFREQ / C,Y,HFREQ / V,N,NØM2PP / V,N,NØB2PP / V,N,NØK2PP /  
      V,N,NØNCUP / V,N,FMØDE / C,Y,KDAMP $
```

#### 4.66.4 Input Data Blocks

USETD - Displacement set definitions table dynamics.

PHIA - Eigenvectors matrix giving the eigenvectors (displacements) in the a set.

MI - Modal mass matrix.

LAMA - Real Eigenvalue Table.

DIT - Direct Input Table.

M2DD - Direct input mass matrix - d set.

B2DD - Direct input damping matrix - d set.

K2DD - Direct input stiffness matrix - d set.

CASECC - Case Control Data Table.

#### Notes:

1. USETD may be purged if  $NØUE < 0$ .
2. PHIA cannot be purged.
3. MI may be purged.
4. LAMA cannot be purged
5. DIT cannot be purged if  $SDAMP \neq 0$  in CASECC.
6. CASECC cannot be purged
7. M2DD cannot be purged if  $NØM2PP \geq 0$ .
8. B2DD cannot be purged if  $NØB2PP \geq 0$ .
9. K2DD cannot be purged if  $NØK2PP \geq 0$ .

## MODULE FUNCTIONAL DESCRIPTIONS

### 4.66.5 Output Data Blocks

- MHH - Modal mass matrix - h set.
- BHH - Modal damping matrix - h set.
- KHH - Modal stiffness matrix - h set.
- PHIDH - Transformation matrix from d set to modal coordinates.

Note: No output matrix can be purged.

### 4.66.6 Parameters

- NØUE - Input-integer-no default. NØUE indicates presence and number of extra points.
- LMØDES - Input-integer-no default. LMØDES selects the first LMØDES eigenvectors (or all if there are less than LMØDES) to use for the modal coordinates.
- LFREQ - Input-real-no default. If LMØDES = 0, eigenvectors with eigenvalues between LFREQ and HFREQ are used in the modal formulation.
- HFREQ - Input-real-no default. See LFREQ.
- NØM2PP - Input-integer-no default. If NØM2PP < 0, M2DD will not be used.
- NØB2PP - Input-integer-no default. If NØB2PP < 0, B2DD will not be used.
- NØK2PP - Input-integer-no default. If NØK2PP < 0, K2DD will not be used.
- NØNCUP - Output-integer-no default. If no direct input matrices exist the problem is considered uncoupled and NØNCUP is set to -1.
- FMØDE - Output-integer-default = 1. The mode number of the first selected eigenvector is stored in FMØDE.
- KDAMP - Input-integer-default = -1. KDAMP chooses the method of computing damping.

### 4.66.7 Method

The general system assembly module for the modal method is used when the real eigenvalues for the structure have been determined. With this method, it is possible to decrease the order of the problem without sacrificing accuracy. The module forms the conversion matrix between modal displacements and all free physical displacements of the system. It then forms the general matrices in terms of displacements of the modes and the extra points.

CASECC is read, and the selected structural damping table "id" is stored.

LAMA is read and the selected eigenvalues are stored in core. If an eigenvalue is selected, the corresponding column of PHIA is copied onto PHIDH1, a scratch file.



# FUNCTIONAL MODULE GKAM (GENERAL K ASSEMBLER MODAL)

If extra points are not present (NØUE < 0), PHIDH = PHIDH1. If extra points are present:

$$[\Phi_{dh}] = \begin{bmatrix} \Phi_a & 1 & 0 \\ - & - & - \\ 0 & 1 & 1 \end{bmatrix} \quad (1)$$

This is accomplished in subroutine GKAM1B.

The "H" matrices are formed:

$$[M_{hh}] = \begin{bmatrix} m_i & 1 & 0 \\ - & - & - \\ 0 & 1 & 0 \end{bmatrix} + [\Phi_{dh}]^T [M_{dd}^2] [\Phi_{dh}] , \quad (2)$$

$$[B_{hh}] = \begin{bmatrix} b_i & 1 & 0 \\ - & - & - \\ 0 & 1 & 0 \end{bmatrix} + [\Phi_{dh}]^T [B_{dd}^2] [\Phi_{dh}] , \quad (3)$$

$$[K_{hh}] = \begin{bmatrix} k_i & 1 & 0 \\ - & - & - \\ 0 & 1 & 0 \end{bmatrix} + [\Phi_{dh}]^T [K_{dd}^2] [\Phi_{dh}] , \quad (4)$$

where  $m_i$  = diagonal terms of MI, and

$$b_i = m_i \omega_i g(\omega_i) , \quad (5)$$

$$k_i = m_i \omega_i^2 , \quad (6)$$

if KDAMP = -1 (the default).

If KDAMP = 1 (used for Aeroelastic),

$$b_i = 0 , \quad (7)$$

$$k_i = (1 + ig(\omega_i)) \omega_i^2 m_i . \quad (8)$$

$\omega_i$  is the frequency for the mode from LAMA and  $g(\omega_i)$  is the tabular structural damping table selected in CASECC. If no selection is made,  $g(\omega_i) = 0.0$ . The "H" matrices are formed using subroutines GKAM1A, SSG2B, TAB, CALCV, MERGE.

## 4.66.8 Subroutines

### 4.66.8.1 Subroutine Name: GKAM1B.

1. Entry Point: GKAM1B.
2. Purpose: To construct  $[\Phi_{dh}]$  if extra points are present.

## MODULE FUNCTIONAL DESCRIPTIONS

3. Calling Sequence: CALL GKAM1B (USETD,SCR1,SCR2,PHIDH,PHIDH1,MØDES,CØRE,LHSET,NØUE)

USETD - GINØ file number of USETD - integer - input.  
 SCR1 - GINØ file number of 1st scratch file - integer - input.  
 SCR2 - GINØ file number of 2nd scratch file - integer - input.  
 PHIDH - GINØ file number of PHIDH - integer - input.  
 PHIDH1 - GINØ file number of PHIDH1 - integer - input.  
 MØDES - Number of modes selected - integer - input.  
 CØRE - Array of open core.  
 LHSET - Length of h set - integer - output.  
 NØUE - Extra point flag NØUE  $\geq 0$  indicates presence of extra points - integer - input.

### 4.66.8.2 Subroutine Name: GKAM1A.

1. Entry Point: GKAM1A.  
 2. Purpose: To form  $[M_{hh}]$ ,  $[B_{hh}]$ , or  $[K_{hh}]$ .  
 3. Calling Sequence: CALL GKAM1A (MI,PHIDH,DIT,SCR1,SCR2,IØPT,IHH,NØI2DD,CØRE,MØDES,SDITD,LHSET,I2DD,IMSKIP,SCR3)

MI - GINØ file number of MI - integer - input.  
 PHIDH - GINØ file number of PHIDH - integer - input.  
 DIT - GINØ file number of DIT - integer - input.  
 SCR1 - GINØ file number of scratch 1 - integer - input.  
 SCR2 - GINØ file number of scratch 2 - integer - input.  
 SCR3 - GINØ file number of scratch 3 - integer - input.  
 IHH - GINØ file number of HH file (M, B, or K) being constructed - integer - input.  
 I2DD - GINØ file number of 2DD file being used with IHH (K2DD, M2DD or B2DD) - integer - input.  
 IØPT - Flag for equation to use

1  $\Rightarrow$  MHH  
 2  $\Rightarrow$  BHH  
 3  $\Rightarrow$  KHH



## FUNCTIONAL MODULE GKAM (GENERAL K ASSEMBLER MODAL)

- integer - input.

NØI2DD - NØI2DD < 0 implies I2DD purged - integer - input.

MØDES - Number of modes selected - integer - input.

SDTID - Id of structural damping table to be used for BHH - integer - input.

LHSET - Length of H set - integer - input.

IMSKIP - Number of records to skip in MI before extracting diagonal terms - integer - input.

CØRE - Array of modes selected.

### 4.66.9 Design Requirements

Three scratch files are necessary. Open core at /GKAM1X/ is used for mode storage. One packed eigenvector must be held in core.

### 4.66.10 Diagnostic Messages

Fatal error messages 3007 and 3008 may be issued by GKAM.



## FUNCTIONAL MODULE DDR1 (DYNAMIC DATA RECOVERY - PART 1)

### 4.67 FUNCTIONAL MODULE DDR1 (DYNAMIC DATA RECOVERY - PART 1)

4.67.1 Entry Point: DDR1

4.67.2 Purpose: To transform modal solutions to physical solutions:

$$\{u_d\} = [\Phi_{dh}] \{u_h\} . \quad (1)$$

### 4.67.3 DMAP Calling Sequence

DDR1 UHV,PHIDH/UDV \$

### 4.67.4 Input Data Blocks

UHV - Solution set displacement vectors.  
PHIDH - Transformation matrix from d set to modal coordinates.

### 4.67.5 Output Data Blocks

UDV - Displacement vectors - d set.

### 4.67.6 Parameters

None

### 4.67.7 Method

Subroutine SSG2B is called to compute  $\{u_d\}$  as in Equation 1.

### 4.67.8 Subroutines

DDR1 has no auxiliary subroutines. See section 3.5.13 for a description of SSG2B.

### 4.67.9 Design Requirements

One scratch file is needed.

## FUNCTIONAL MODULE DDR2 (DYNAMIC DATA RECOVERY - PART 2)

### 4.68 FUNCTIONAL MODULE DDR2 (DYNAMIC DATA RECOVERY - PART 2)

#### 4.68.1 Entry Point: DDR2

#### 4.68.2 Purpose

To compute mode acceleration displacements.

#### 4.68.3 DMAP Calling Sequence

DDR2    USETD,UDV,PDF,K2DD,B2DD,MDD,FRL,ULL,LLL,DM/UDV1,UEVF,PAF/V,N,TYPE/V,N,NØUE/V,N,REACT/  
         V,N,FRQSET    \$

#### 4.68.4 Input Data Blocks

USETD    - Displacement set definitions table dynamics.  
UDV       - Displacement vectors - d set.  
PDF       - Dynamic load matrix for frequency analysis - d set.  
K2DD      - Direct input stiffness matrix - d set.  
B2DD      - Direct input damping matrix - d set.  
MDD       - Dynamic mass matrix - d set.  
FRL       - Frequency Response List.  
ULL       - Upper triangular factor of KLL -  $\ell$  set.  
LLL       - Lower triangular factor of KLL -  $\ell$  set.  
DM        - Rigid body transformation matrix.

#### Notes:

1. USETD must not be purged.
2. UDV must not be purged.
3. PDF must not be purged.
4. FRL must not be purged if TYPE = FREQ.
5. MDD must not be purged.
6. ULL, LLL must not be purged.
7. DM must not be purged if REACT  $\geq$  0.

## MODULE FUNCTIONAL DESCRIPTIONS

### 4.68.5 Output Data Blocks

- UDV1 - Displacements after mode acceleration - d set.
- UEVF - Displacements at the extra points.
- PAF - Equivalent load vector for mode acceleration computations - a set.

### 4.68.6 Parameters

- TYPE - Input-BCD-no default. TYPE determines the type of mode acceleration which will be used, TRAN for transient or FREQ for frequency response.
- NØUE - Input-integer-no default.  $NØUE \geq 0$  indicates presence of extra points.
- REACT - Input-integer-no default.  $REACT \geq 0$  indicates presence of supports.
- FRQSET - Input-integer-no default. FRQSET chooses the frequency list if TYPE = FREQ.

### 4.68.7 Method

The equivalent load vector is computed:

$$\{P_d^a\} = \{P_d\} - [K_{dd}^2] \{u_d\} - [B_{dd}^2] \{\dot{u}_d\} - [M_{dd}] \{\ddot{u}_d\} . \quad (1)$$

For a transient analysis problem  $\{u_d\}$ ,  $\{\dot{u}_d\}$ , and  $\{\ddot{u}_d\}$  are given explicitly. For Frequency Response Analysis:

$$\{\dot{u}_d\} = i\omega \{u_d\} , \quad (2)$$

$$\{\ddot{u}_d\} = -\omega^2 \{u_d\} , \quad (3)$$

where  $\omega$  is the forcing frequency and  $\{u_d\}$  is the complex response vector.  $\omega$  comes from FRQSET in FRL. The vector  $\{P_d^a\}$  is the sum of applied loads and inertia loads due to the motion of the system approximated by its lower modes. The static solution using these loads will provide a better answer for displacements.



# FUNCTIONAL MODULE DDR2 (DYANMIC DATA RECOVERY - PART 2)

If extra points are present ( $N\emptyset UE \geq 0$ ), then

$$\{P_d^a\} \Rightarrow \left\{ \frac{P_a^a}{P_e} \right\}, \quad (4)$$

$$\{u_d\} \Rightarrow \left\{ \frac{u_a^a}{u_e} \right\}. \quad (5)$$

$\{u_e\}$  is placed in data block UEVF. Subroutines CALCV and SSG2A perform this calculation.

If supports are present ( $REACT \geq 0$ ), then

$$\{P_\ell^a\} \Rightarrow \left\{ \frac{P_\ell}{P_r} \right\}, \quad (6)$$

$$\{u_a\} \Rightarrow \left\{ \frac{u_\ell}{u_r} \right\}. \quad (7)$$

Solve for  $\{u_a^a\}$ :

$$[L_{\ell\ell}] [U_{\ell\ell}] \{u_\ell^a\} = \{P_\ell^a\}. \quad (8)$$

This is accomplished in subroutine SSG3A.

If supports are present, then

$$\{u_a^a\} = \left\{ \frac{\{u_\ell^a\} + [D] \{u_r\}}{u_r} \right\}, \quad (9)$$

otherwise,  $\{u_a^a\} = \{u_\ell^a\}$ . Subroutine SDR1B performs this calculation.

## MODULE FUNCTIONAL DESCRIPTIONS

If extra points are present, then

$$\{u_d^a\} \Leftarrow \begin{Bmatrix} u_a^a \\ \hline u_e \end{Bmatrix}. \quad (10)$$

Note: If the problem type is transient,  $\{u_d^a\}$  must be merged with  $\{\dot{u}_d\}$  and  $\{\ddot{u}_d\}$ .

### 4.68.8 Subroutines Called

CALCV - See section 3.5.5.

SSG2A - See section 3.5.7.

SSG2B - See section 3.5.13.

SSG3A - See section 3.5.18.

SDR1B - See section 3.5.8.

#### 4.68.8.1 Subroutine Name: DDR1A.

1. Entry Point: DDR1A.
2. Purpose: To construct the equivalent load vector  $\{p_d^a\}$ .
3. Calling Sequence: CALL DDR1A(PDF,K2DD,B2DD,MDD,UDV,PAF,FRL,FRQSET,SCR1,SCR2,SCR3,SCR4,TYPE,SCR5).

|   |   |   |
|---|---|---|
| PDF<br>K2DD<br>B2DD<br>MDD<br>UDV<br>PAF<br>FRL<br>SCR1-5 | } | GINØ file number of appropriate data block - integer - input. |
|---|---|---|

FRQSET - Frequency set list id - integer - input. FRQSET will be used only if TYPE = FREQ.

TYPE - Problem type - BCD - input.

## FUNCTIONAL MODULE DDR2 (DYNAMIC DATA RECOVERY - PART 2)

### 4.68.8.2 Subroutine Name: DDR1B

1. Entry Point: DDR1B.
2. Purpose: To merge displacements with previously computed velocity and acceleration in a transient problem.
3. Calling Sequence: CALL DDR1B (UDV,UAD,UADV).

UDV - GINØ file number of displacement, velocity and acceleration file - integer - input.

UAD - GINØ file number of equivalent displacements - integer - input.

UADV - GINØ file number of new displacements, velocity and acceleration - integer - input.

### 4.68.9 Design Requirements

Open core for DDR2 begins at /DDR1X/. Open core for DDR1A begins /DDRA1/. Open core for DDR1B begins /DDRB1/. Six scratch files are needed.

### 4.68.10 Diagnostic Messages

None.



## OUTPUT MODULE XYPLØT (X-Y DATA PLOTTER)

### 4.69 OUTPUT MODULE XYPLØT (X-Y DATA PLOTTER)

#### 4.69.1 Entry Point: XYPLØT

#### 4.69.2 Purpose

To process information supplied by module XYTRAN through a single data block and output to either PLT1 (BCD plot tape) or PLT2 (binary plot tape) for labeling and plotting X-Y data on an off-line plotter.

#### 4.69.3 DMAP Calling Sequence

XYPLØT      XYPLTT// \$

#### 4.69.4 Input Data Blocks

XYPLTT    -   Plotting Control Values Table. Note if XYPLTT is purged, XYPLØT returns control without action.

#### 4.69.5 Output Data Blocks

None. (All output consists of physical tapes produced for off-line plotters and possibly user warning messages to the installation output unit for printing).

#### 4.69.6 Parameters

None.

#### 4.69.7 Method

XYPLØT initially determines open core size and assigns buffers for its input file and output file. The remaining core is used to store data points read in for each plot. The input file is then opened and spaced forward over the header record containing the data block name. Should the system not be able to locate this file, a warning message is output and XYPLØT returns control to the calling program without further action. Otherwise XYPLØT reads in the first I.D. record from the input file. A check is made to determine if the word count of this record is correct. If not, the following records are checked until either the correct

## MODULE FUNCTIONAL DESCRIPTIONS

word count is found or the error count reaches a specified limit. If the specified limit is reached, XYPLØT assumes the input file is invalid and returns control to the calling program after printing a warning message.

If the I.D. record had the proper word count, XYPLØT checks if new axes are necessary. If not, the next data record is read, and the data pairs are plotted on the previous axes. When new axes are necessary, a check is made to determine if they go on the lower half of a plot. If not, XYPLØT makes a number of I.D. data validity checks. Whenever possible, where I.D. data are questionable, default values are assigned and processing continues following a warning message that this particular plot may be invalid.

After the validity checks, XYPLØT terminates the previous plot and initializes the plotting parameters for the NASTRAN plotting software. This is done for each new plot so that it is possible to produce alternate plots on two different plotters. Normally, however, plots will be done for only one plotter on any single entry to XYPLØT. If required, a new plot is initiated, and curve and axes titles are prepared from the I.D. data and generated. If not a new plot, only the axes titles are done.

At this time XYPLØT computes the constants which will be used to transform the curve data into actual plotter counts. These constants are saved and used until new axes are drawn.

Following this, XYPLØT determines if any tick marks are to be placed along the X axis and at the X maximum and X minimum lines. If there are to be tick marks, the number and spacing (linear or logarithmic) is computed for them and plotted. As the X direction tick marks are prepared, a check is made to determine if Y grid lines are requested. If so, a grid line is prepared at each tick mark and plotted. Tick mark labels are prepared and plotted at the same time as the tick marks and grid lines, if any.

After the tick marks are completed, the X and Y axes are plotted if requested.

Once the curve titles, tick marks, and labeling have been accomplished, XYPLØT reads in the next record from the input data file. Normally all the data pairs for any I.D. record can be brought into core memory with a single read. However, provision is made for additional reads if the open core space is not sufficient to contain all the data on the initial read. A check is made to determine if there are an even number of data values (i.e., an X and Y value for each data



## OUTPUT MODULE XYPLØT (X-Y DATA PLOTTER)

point). If not, a warning message is printed and the last value ignored. The data are then checked against the previously defined X and Y frame minimums and maximums (integer one for the X value means skip the point). Any data outside these limits are ignored and not plotted. The remaining data points are then converted to plotter counts and plotted in one of three modes. The three modes are: point plot with choice of symbol; line plot; combination of the first two.

After finishing the data, XYPLØT reads in the next I.D. record and continues as before until and end-of-file is reached. At this point it closes the input file, terminates the current plot and returns control to the calling program.

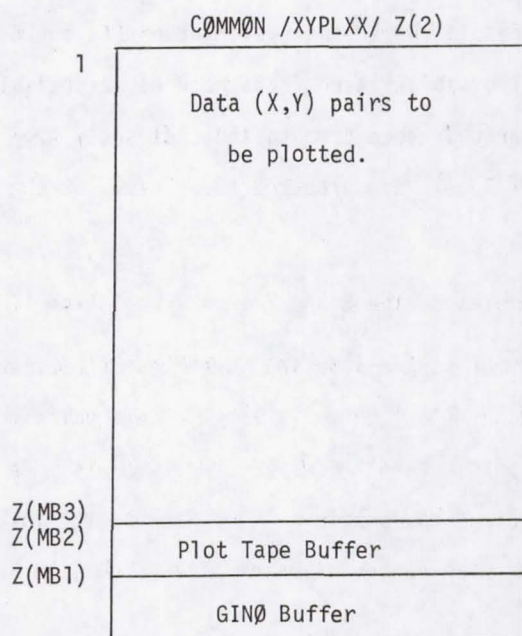
### 4.69.8 Subroutines

XYPLØT calls the following plotter utility subroutines: AXIS, LINE, PRINT, SØPEN, PLTSET, STPLØT, SYMBØL, TIPE, TYPFLT, and TYPINT. The descriptions of these subroutines may be found in Section 3.4,

### 4.69.9 Design Requirements

#### 4.69.9.1 Allocation of Core Storage

XYPLØT uses open core for two GINØ buffers and the remainder as one large buffer for data points. It appears as follows:





## MODULE FUNCTIONAL DESCRIPTIONS

Normally the data pairs buffer will be sufficiently large to hold all the data pairs for a single curve at one time. However, this is not necessary and XYPLØT could operate if the data pairs buffer were only two words long, although not efficiently. As an output module, XYPLØT has been programmed to avoid any system fatal errors. The worst condition that should occur is that no plots are produced. In all cases XYPLØT returns to the calling program so that other system functions may be continued.

### 4.69.9.2 Environment

The beginning of open core for XYPLØT is defined by /XYPLXX/. XYPLØT uses no scratch files. Common storage requirements consist of /XXPARM/ and /PLTDAT/ which are defined in the block data deck PLØTBD which must be loaded with XYPLØT. /CHAR94/ and SYMBLS/ are also defined in PLØTBD and are necessary for the subroutines called by XYPLØT. See Section 2.5 for a description of these common blocks.

When XYPLØT is called, there must be at least one physical tape set up to receive the plotted output, otherwise XYPLØT returns to the calling program without further action.

### 4.69.10 Diagnostic Messages

Diagnostic messages 991 through 997 may be output on the installation printer device as a result of XYPLØT operation. Generally they are self-explanatory and usually point out particular plots which are questionable rather than giving the user a precise method of solving the problem. This is not possible since XYPLØT receives all its information through a series of other modules rather than from the user directly. See Section 6 of the User's Manual for details.

## OUTPUT MODULE ØFP (OUTPUT FILE PROCESSOR)

### 4.70 OUTPUT MODULE ØFP (OUTPUT FILE PROCESSOR)

#### 4.70.1 Entry Point: ØFP

#### 4.70.2 Purpose

ØFP outputs to the system output file, in user-oriented, self-explanatory formats, data blocks prepared for output by other functional modules.

#### 4.70.3 DMAP Calling Sequence

ØFP DB1,DB2,DB3,DB4,DB5,DB6//V,N,CARDNØ \$

#### 4.70.4 Input Data Blocks

One to six input data blocks in the output order desired. Any or all input data blocks may be purged.

#### 4.70.5 Output Data Blocks

None

#### 4.70.6 Parameters

CARDNØ - Input and output - integer - default = 0. CARDNØ is incremented by one and punched in columns 73-80 for each card punched by ØFP.

#### 4.70.7 Method

##### 4.70.7.1 Overall Logic Flow

The ØFP logic consists of defining one GINØ buffer and then entering one overall loop of six passes (one pass for each data block). All input data blocks are then handled identically one at a time.

Within each data block, each odd numbered (Identification) record and its respective immediately following even numbered (Data) record are considered as a pair, and is a completely separate entity. There is, and need be, no correspondence between these two records and the previous two records, or between these two records and the following two records.

## MODULE FUNCTIONAL DESCRIPTIONS

Thus, within the loop for a given data block, after the file on which the data block resides is opened and its header record is skipped, ØFP reads an Identification record, defines various pointers and descriptors, and then, if any data are present in the Data record, processes this data line by line until the end-of-record is reached. This process continues for all Identification-Data record pairs.

### 4.70.7.2 Defining Descriptors and Pointers

Because ØFP was confronted with outputting a vast array of data classes having many data format and heading format configurations, it was decided that in order to keep ØFP from becoming a mammoth module of format statements, a system of pointers would be used in conjunction with all the different micro-format elements required.

Information in the Identification record is sufficient to select an initial class pointer. This class pointer, with the addition of a subclass pointer, points to an array of six pointers, five of which define five micro-line formats (from the master set of micro-line formats), and one of which points to a string of micro-data format pointers. These micro-data format pointers then each point to a micro-data format capable of outputting a single variable.

This design is such as to make possible the definition of macro-formats and to allow for easy modification and addition of more output data classes.

### 4.70.8 Subroutines

#### 4.70.8.1 Subroutine Name: ØFPPUN

1. Entry Point: ØFPPUN
2. Purpose: To write output on the system punch unit.
3. Calling Sequence: CALL ØFPPUN (BUF,NWDS,IØPT,IDD,PNCHED)  
BUF        - Array to be output.  
NWDS       - Number of words in BUF to output.  
IØPT       - { 1 = Vector output.  
              2 = General output.  
IDD        - { 0 = SØRT1 (1st word Integer).  
              1 = SØRT2 (1st word Real).



## OUTPUT MODULE ØFP (OUTPUT FILE PROCESSOR)

PNCHED -  $\begin{cases} \text{.FALSE.} = \text{Punch heading cards.} \\ \text{.TRUE.} = \text{Do not punch heading cards.} \end{cases}$

### 4.70.8.2 Subroutine Name: ØFP1

1. Entry Point: ØFP1
2. Purpose: To call PAGE and write five micro-line formats.
3. Calling Sequence: CALL ØFP1

### 4.70.8.3 Subroutine Name: ØFP1A

1. Entry Point: ØFP1A
2. Purpose: An auxiliary routine to ØFP1. Called by ØFP1 only.
3. Calling Sequence: CALL ØFP1A(LINE)  
LINE - Integer - Branch to format pointer.

### 4.70.8.4 Block Data Subprogram Name: ØFP1BD

ØFP1BD defines common block /ØFPBD1/.

### 4.70.8.5 Block Data Subprogram Name: ØFP2BD

ØFP2BD defines common block /ØFPBD2/.

### 4.70.8.6 Block Data Subprogram Name: ØFP3BD

ØFP3BD defines common block /ØFPBD3/.

### 4.70.8.7 Block Data Subprogram Name: ØFP4BD

ØFP4BD defines common block /ØFPBD4/.

### 4.70.8.8 Block Data Subprogram Name: ØFP5BD

ØFP5BD defines common block /ØFPBD5/.

## MODULE FUNCTIONAL DESCRIPTIONS

### 4.70.9 Design Requirements

The common blocks listed above interface between the main subroutines ØFP and ØFP1A. In addition COMMON/ØFPXXX/ is used to define open-core which contains the following.

|                            |   |  |
|----------------------------|---|--|
| L1<br>L2<br>L3<br>L4<br>L5 | } | Five words which indicate the five format numbers defining the heading for the current data being output.              |
| ID                         | - | A fifty word buffer for storage of the first fifty words of an identification record from the data block to be output. |
| BUFF                       | - | A GINØ buffer.   |

The pointer system required by the design operates as described below. The arrays B,C,D,E, and ESINGL, referenced in the discussion below appear in subroutine ØFP.

1. The variable I is set equal to the Data type specified in the data block. The variable J is set equal to the class of data: 1 = Real - SØRT1, 2 = Complex - SØRT1, etc. Then the base pointer, CPØINT = B(I,J), is found.

Data type 1

Data type 2

.

.

.

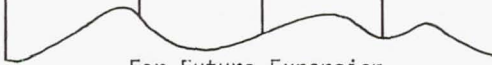
.

.

Data type N

B array

| SØRT 1 |         | SØRT 2 |         |
|--------|---------|--------|---------|
| Real   | Complex | Real   | Complex |
| 0      | 130     | 120    | 132     |
| 2      | 134     | 122    | 136     |
| 4      | 138     | 124    | 140     |
| .      | .       | .      | .       |
| .      | .       | .      | .       |
| .      | .       | .      | .       |
| .      | .       | .      | .       |
| .      | .       | .      | .       |
| .      | .       | .      | .       |
| .      | .       | .      | .       |
| .      | .       | .      | .       |
| .      | .       | .      | .       |
| .      | .       | .      | .       |



For Future Expansion

Example:  
For I = 2, J = 4.

CPØINT = B(I,J)  
= 136

Example:  
For I = 2, J = 4.

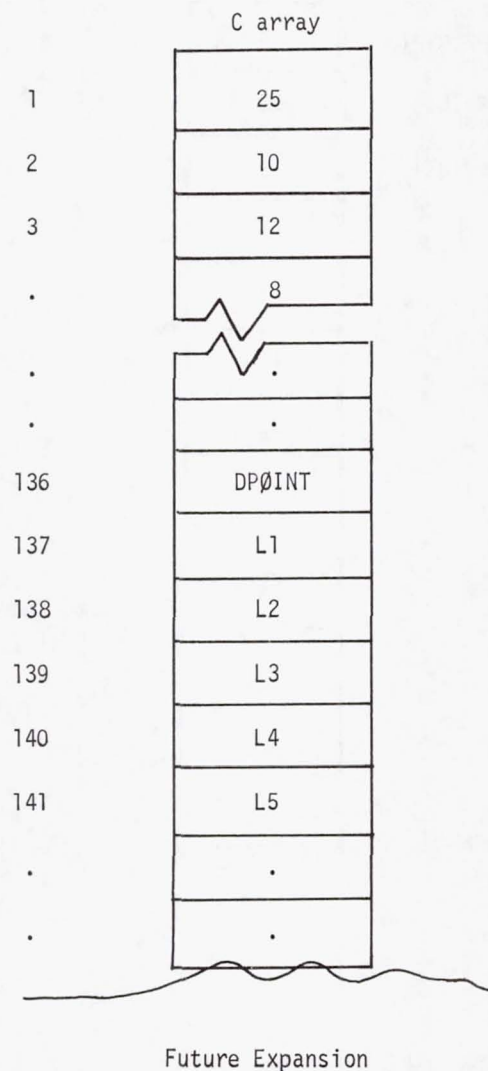
CPØINT = B(I,J)  
          = 136

# OUTPUT MODULE ØFP (OUTPUT FILE PROCESSOR)

2. CPØINT is a index into the C array. Define: DPØINT = C(CPØINT). DPØINT is an index into the D array. Also

$L1 = C(CPØINT + 1),$   
 $L2 = C(CPØINT + 2),$   
 $L3 = C(CPØINT + 3),$   
 $L4 = C(CPØINT + 4),$   
 and  $L5 = C(CPØINT + 5).$

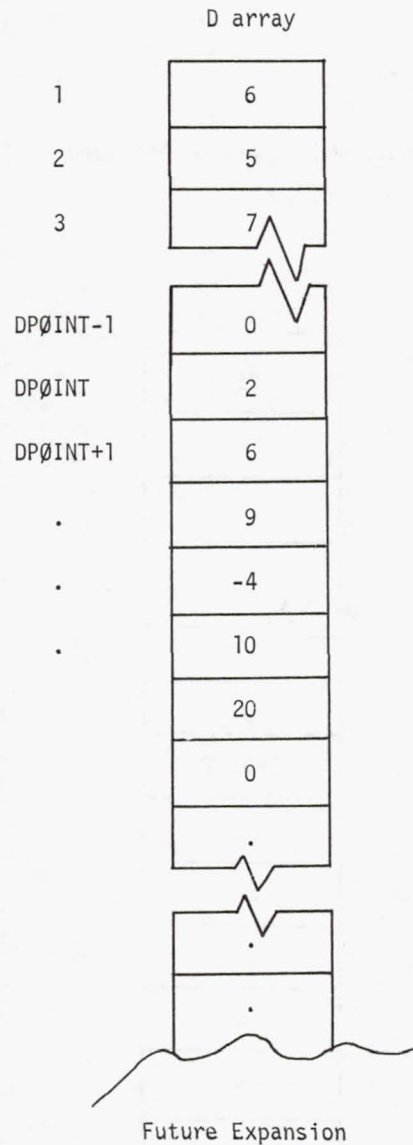
These are the 5 line format numbers which make up the heading format for the type of data currently being processed.





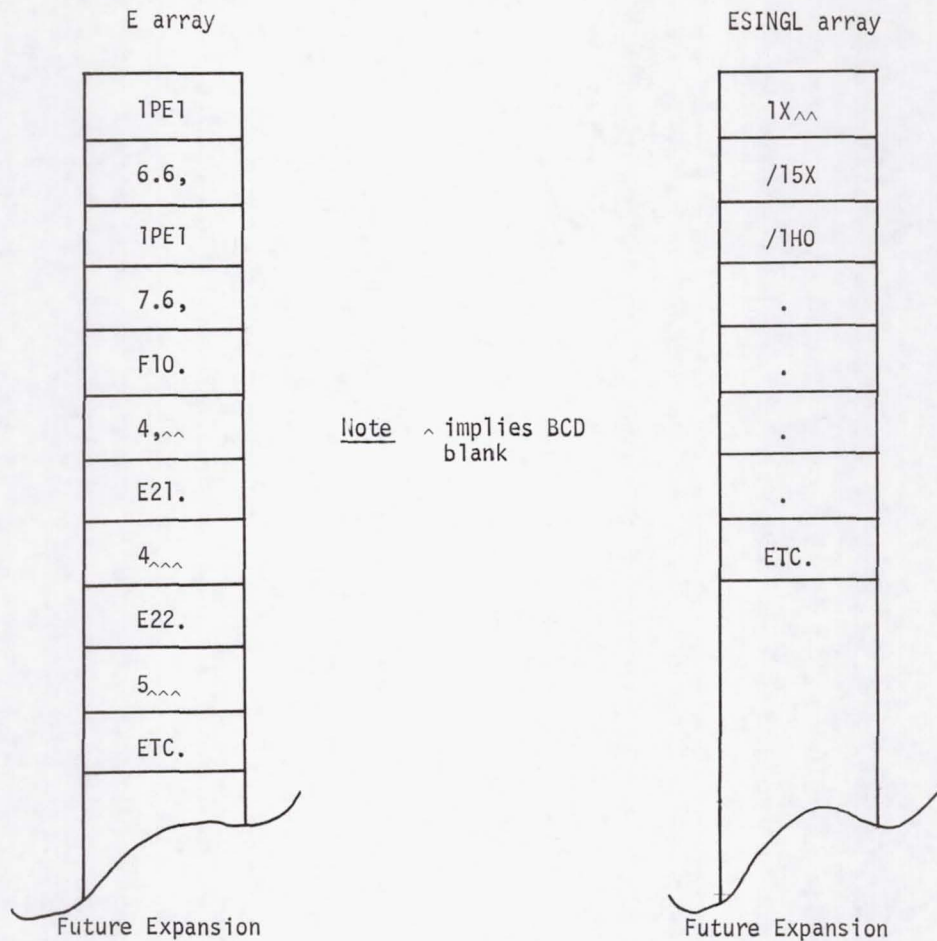
# MODULE FUNCTIONAL DESCRIPTIONS

3. Word DPØINT of the D array defines the beginning of a string of pointers into the E array. This string is terminated by the first word containing a zero. Each word of this string thus defines a string of words in the E array which contains Hollerith data for construction of a format. Should a word in the D array be negative, the absolute value is used to point into the ESINGL array which contains Hollerith data also.



## OUTPUT MODULE ØFP (OUTPUT FILE PROCESSOR)

4. The E array contains Hollerith data pertaining to the output of a variable; the ESINGL array contains Hollerith data pertaining to spacing and carriage control only.



### 4.70.10 Diagnostic Messages

If, during some phase of outputting a data block, ØFP encounters an error condition, work on that data block will cease, a warning message will be printed, and a call to the NASTRAN table-printer for table printing of this data block will be made. ØFP will then continue processing the remaining input data blocks.

## OUTPUT MODULE MATPRN (GENERAL MATRIX PRINTER)

### 4.71 OUTPUT MODULE MATPRN (GENERAL MATRIX PRINTER)

#### 4.71.1 Entry Point: MATPRN.

#### 4.71.2 Purpose

To print general matrix data blocks.

#### 4.71.3 DMAP Calling Sequence

MATPRN KGG,PL,PG,B2PP,UPV// \$

#### 4.71.4 Input Data Blocks

KGG - Any matrix data block.

PL - Any matrix data block.

PG - Any matrix data block.

B2PP - Any matrix data block.

UPV - Any matrix data block.

#### Notes:

1. Any or all input data blocks can be purged.
2. If any data block is not a matrix, the TABPT routine will be called.

#### 4.71.5 Output

The non-zero band of each column of the input matrix is unpacked and is printed in single precision format on the system output file.

#### 4.71.6 Parameters

None.

#### 4.71.7 Method

Subroutine MATDUM is called for each non-purged input file.

#### 4.71.8 Subroutines

MATDUM - See subroutine description, section 3.4.28.



#### 4.72 OUTPUT MODULE MATGPR (DISPLACEMENT METHOD MATRIX PRINTER)

##### 4.72.1 Entry Point: MATGPR

##### 4.72.2 Purpose

To print displacement method matrices, identifying values with external grid point numbers.

##### 4.72.3 DMAP Calling Sequence

MATGPR GPL,USET,SIL,ANYMAT//C,N,CØLSET/C,N,RØWSET \$

##### 4.72.4 Input Data Blocks

GPL - Grid Point List (This may also be GPLD if extra points are present.)

USET - Displacement set definitions table (This may also be USETD if extra points are present.)

SIL - Scalar Index List (This may also be SILD if extra points are present.)

ANYMAT- Any displacement method matrix.

##### Notes:

1. Unless CØLSET = RØWSET = 'H', GPL, USET and SIL must be present.
2. If ANYMAT is purged, MATGPR will return.

##### 4.72.5 Output Data Blocks

The non-zero terms of ANYMAT are given external identification and printed on the system output file.

##### 4.72.6 Parameters

CØLSET - Input-BCD-no default. CØLSET indicates the set to which the columns of ANYMAT belong. If CØLSET is not one of the following: M,Ø,R,SG,SB,L,A,F,S,N,G,E,P,NE,FE,D,H then MATGPR will return.

RØWSET - Input-BCD-default = X. RØWSET indicates the set to which the rows of ANYMAT belong. If RØWSET is not a legal set name, RØWSET = CØLSET.

## MODULE FUNCTIONAL DESCRIPTIONS

### 4.72.7 Method

The BCD parameters CØLSET and RØWSET are converted to bit positions in USET. CØLSET must be one of the following 17 symbols: M,Ø,R,SG,SB,L,A,F,S,N,G,E,P,NE,FE,D,H or else MATGPR will return. If ROWSET is not a legitimate symbol RØWSET = CØLSET.

GPL, USET, and SIL are placed in core. Each column and non-zero row element is identified according to the following scheme:

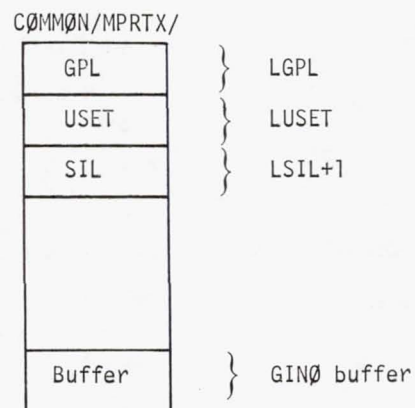
1. USET is searched for the number of members belonging to the g set (p set if USETD is used) before the current member of the matrix set.
2. This number is looked up in SIL to obtain the internal grid point number and type of point (scalar, grid, or extra).
3. The internal grid point number points into GPL for the external ID.

### 4.72.8 Subroutines

MATGPR has no auxiliary subroutines.

### 4.72.9 Design Requirements

1. Open core is defined at /MPRTX/.
2. Layout of open core is as follows:



OUTPUT MODULE MATGPR (DISPLACEMENT METHOD MATRIX PRINTER)

4.72.10 Diagnostic Messages

MATGPR may issue the following diagnostic messages:

3007 and 3008.



## OUTPUT MODULE MATPRT (MATRIX PRINTER)

### 4.73 OUTPUT MODULE MATPRT (MATRIX PRINTER)

#### 4.73.1 Entry Point: PRTINT

#### 4.73.2 Purpose

To print a matrix data block.

#### 4.73.3 DMAP Calling Sequence

MATPRT X//C,N,RC/C,N,Y \$

#### 4.73.4 Input Data Block

X - Matrix data block to be printed. If X is purged, then nothing is done.

#### 4.73.5 Output Data Blocks

None.

#### 4.73.6 Parameters

RC - Indicates whether X is stored by rows (RC = 1) or columns (RC = 0) - integer - input.

Y - Indicates whether X is to be printed (Y < 0, do not print X; Y > 0, print X)  
- integer - input - default value = -1.

#### 4.73.7 Method

Each column (or row) of the matrix is broken into groups of 6 terms (3 terms if complex) per printed line. If all the terms in a group = 0, the line is not printed. If the entire column (or row) = 0, it is not printed. If the entire matrix = 0, it is not printed.

#### 4.73.8 Subroutines

##### 4.73.8.1 Subroutine Name: INTPRT

1. Entry Point: INTPRT
2. Purpose: To print a matrix data block using subroutine MATPRT.
3. Calling Sequence: CALL INTPRT (A,CR,Ø,NAME)

## MODULE FUNCTIONAL DESCRIPTIONS

where:

- A - Storage for 1 column (row) of the matrix + 1 GINØ buffer.
- CR - { 0 if the matrix is stored by columns.  
1 if the matrix is stored by rows.
- Ø - { 0 if the matrix is not to be printed.  
1 if the matrix is to be printed.
- NAME - 8 character name of the matrix (2 words, 4 characters per word).

3. Method: Subroutine MATPRT is called to print the matrix. Whenever MATPRT returns for a matrix name or column/row id to be printed, the name of the matrix (NAME<sub>1</sub>, NAME<sub>2</sub>) or the column or row id (as indicated by 'CR'), is printed.

### 4.73.8.2 Subroutine Name: MATPRT

1. Entry Points: MATPRT, PRTMAT
2. Purpose: To print a matrix data block.
3. Calling Sequence: CALL MATPRT (\$N<sub>1</sub>, \$N<sub>2</sub>, A, ØPT, CØLNUM)  
CALL PRTMAT (\$N<sub>1</sub>, \$N<sub>2</sub>)  
CØMMØN/XXMPRT/MCB(7)

where:

- N<sub>1</sub> - FØRTRAN statement number defining the return executed whenever a new page has been started (the calling program is expected to print the matrix and column id. CØLNUM = current column number).
- N<sub>2</sub> - FØRTRAN statement number defining the return executed whenever the column id must be printed in the middle of a page (CØLNUM = current column number).
- A - Storage for 1 column of the matrix + one GINØ buffer.
- ØPT - See subroutine description for VECPT, below, for the explanation of this argument.
- CØLNUM - Current column number being printed (output).
- MCB - Matrix control block.

## OUTPUT MODULE MATPRT (MATRIX PRINTER)

3. Method: The matrix is unpacked and printed one column at a time. Whenever either of the nonstandard returns ( $\$N_1, \$N_2$ ) is executed, the calling program must call PRTMAT to continue the printing of the matrix.

4. Additional Subroutines Called: VECprt.

### 4.73.8.3 Subroutine Name: VECprt

1. Entry Points: VECprt, PRTVEC
2. Purpose: To print a vector.
3. Calling Sequence: CALL VECprt ( $\$N_1, \$N_2, P, N, A, \emptyset PT$ )  
CALL PRTVEC ( $\$N_1, \$N_2$ )

where:

- $N_1$  - FØRTRAN statement number defining the return executed whenever a new page has been started (the calling program is expected to print the vector id and any other subtitles desired).
- $N_2$  - FØRTRAN statement number defining the return executed whenever the vector id is to be printed in the middle of a page.
- P - Vector type and precision.
- N - Number of components in the vector.
- A - Location of the vector.
- $\emptyset PT$  -  $\left\{ \begin{array}{l} = 0 \text{ if all the vector components are to be printed, regardless of its sparsity, and if it is to be printed starting on a new page if it will not fit on the current page.} \\ = +1 \text{ if only the printed lines which would have at least one non-zero component are to be printed, and if the vector is to be printed starting on a new page if it will not fit on the current page.} \\ = -1 \text{ if only the printed lines which would have at least one non-zero component are to be printed, and if as much of the vector as possible is to be printed on the current page.} \end{array} \right.$

3. Method: The vector will be printed as a single precision real or complex vector. The components will be printed 6 per line if real, 3 per line if complex. In addition, the first and last components of each line will be identified on each side of the line by their respective component members. In addition whenever either of the nonstandard



## MODULE FUNCTIONAL DESCRIPTIONS

returns ( $\$N_1, \$N_2$ ) is executed the calling program must call PRTVEC to continue the printing of the vector.

4. Additional Subroutines Called: FØRMAT.

### 4.73.8.4 Subroutine Name: FØRMAT

1. Entry Point: FØRMAT
2. Purpose: To print a line of 1 to 6 real numbers (optionally centered) preceded and followed by integer id's of the first and last number printed.
3. Calling Sequence: CALL FØRMAT (A,N1,N2,N3,L1,L2)

where:

- A - Array from which the 1 to 6 real numbers are to be printed.
- N1 - Index of the 1st number in the array to be printed.
- N2 - Index of the last number in the array to be printed.
- N3 - Increment to be used in extracting the 2nd, 3rd, etc., numbers in the array to be printed.
- L1 - Integer id of the 1st number to be printed.
- L2 - Integer id of the last number to be printed.

3. Method: If L1 and L2 are both positive, the numbers will be centered on the page. If either L1 or L2 is not positive, the numbers will be printed based upon the centering of 6 numbers.

### 4.73.9 Design Requirements

Open core is defined at /XXPRTI/. Open core contains one GINØ buffer followed by one unpacked real or complex single precision column of the matrix.

## OUTPUT MODULE SEEMAT (PICTORIAL MATRIX PRINTER)

### 4.74 OUTPUT MODULE SEEMAT (PICTORIAL MATRIX PRINTER)

#### 4.74.1 Entry Point: SEEMAT

#### 4.74.2 Purpose

To show nonzero matrix elements on printer or plotter output positioned pictorially by row and column within the outlines of the matrix.

#### 4.74.3 DMAP Calling Sequence

```
SEEMAT  M1,M2,M3,M4,M5//C,N,{PRINT  
        {PLØT}}/V,N,PFILE/V,N,PACK/C,N,PLØTTER/C,N,MØDELN1/C,N,  
        MØDELB1/C,N,MØDELN2/C,N,MØDELB2 $
```

Note that parameters PLØTTER, MØDELN1, MØDELB1, MØDELN2 and MØDELB2 are all described in paragraph 4 in section 4.74.6.

#### 4.74.4 Input Data Blocks

```
M1 )  
M2 )  
M3 ) Matrix Data Blocks, any of which may be purged.  
M4 )  
M5 )
```

#### 4.74.5 Output Data Blocks

None. The formatted matrix picture is output on the system output file or on a plot tape depending on the value of the first parameter.

#### 4.74.6 Parameters

1. PRINT implies use of the system output file. (Any value other than PLØT implies PRINT). PLØT implies use of one of the plotters. Either of the plotter tapes PLT1 or PLT2 will be used, depending on the type of plotter requested.

2. PFILE is the plot number (Used only if first parameter is PLØT).

Input/output variable integer parameter. Frame or sheet number. The value of this parameter will be incremented by one (1) for each frame (sheet) created by SEEMAT.

The default value for this parameter is 0.

3. PACK is reserved for a future modification that will allow the representation of a nonzero block of the matrix with a single character. This parameter may be specified as C,N only. (see example in paragraph 4 below)

# MODULE FUNCTIONAL DESCRIPTIONS

## 4. Plotter Name and Model Identification (Used only if parameter 1 = PLØT.)

Each plotter name has associated with it two model identifiers. Each of these model identifiers may either be an integer (MØDELNi) value or BCD (MØDELBi) value. If model identifier "i" (i = 1, 2) is an integer, insert its value for MØDELNi; if model identifier "i" (i = 1, 2) is BCD, insert its value for MØDELBi. In either case, specify the other value for model identifier "i" (MØDELBi and MØDELNi, respectively) as C,N only.

Below is a list of model identifiers allowable for each plotter name. A detailed explanation of this list can be found in section 4 of the User's Manual. Each plotter has associated with it a default model and several optional models. The model underlined is the default model. To access this default model, do not specify any of the last four DMAP parameters. For example to specify the CALCØMP 765, 205 (see section 4 of the User's Manual) the following DMAP statement may be used:

SEEMAT M1,M2,M3,M4,M5//C,N,PLØT/V,N,PFILE/C,N/C,N,CALCØMP \$

| <u>Plotter Name</u> | <u>Model Identifiers</u>  |
|---------------------|---|
| BL                  | { <u>LTE,30</u> }<br>{ STE,30 }   |
| EAI                 | { <u>3500,300</u> }<br>{ 3500,45 }  |
| SC                  | <u>4020,0</u>   |
| CALCØMP             | ( <u>765,205</u> )<br>( 765,210 )<br>( 765,105 )<br>( 765,110 )<br>( 763,205 )<br>( 763,210 )<br>( 763,105 )<br>( 763,110 )<br>( 565,205 )<br>( 565,210 )<br>( 565,105 )<br>( 565,110 )<br>( 565,305 )<br>( 565,310 )<br>( 563,205 )<br>( 563,210 ) |



# OUTPUT MODULE SEEMAT (PICTORIAL MATRIX PRINTER)

| <u>Plotter Name</u> | <u>Model Identifiers</u>   |
|---------------------|--|
|                     | $\begin{pmatrix} 563,105 \\ 563,110 \\ 563,305 \\ 563,310 \end{pmatrix}$ |
| DD                  | <u>80,B</u>  |
| NASTRAN             | $\begin{pmatrix} M,0 \\ T,0 \\ D,0 \\ M,1 \\ T,1 \\ D,1 \end{pmatrix}$   |

where:

- BL is a Benson-Lehner plotter
- EAI is an Electronic Associates plotter
- SC is a Stromberg Carlson plotter
- CALCOMP is a California Computer plotter
- DD is a Data Display plotter
- NASTRAN is the NASTRAN general purpose plotter

## 4.74.7 Method

The matrix is partitioned into blocks which can be printed on a single sheet of output paper or plotted on a single frame or sheet of plotter output media. Only blocks containing nonzero elements will be printed. Row and column indices are indicated. The user of this module is cautioned to make sure that his line count limit is large enough. A default of 20,000 lines is provided by NASTRAN. This may be changed via the statement "MAXLINES = value" in the NASTRAN Case Control Deck. The transpose of the matrix is always printed.

The columns of the matrix are examined for nonzero terms. Let the matrix be partitioned into blocks, where a block consists of NL columns and 100 rows, where NL is the number of data lines per page obtained from /SYSTEM/. For each block containing nonzero terms, a BCD block image is stored in open core in packed bit format. Only blocks containing nonzero terms are stored. When NL columns have been passed, the blocks containing nonzero terms are printed on

## MODULE FUNCTIONAL DESCRIPTIONS

the system output file or plotted. Note that since NASTRAN matrices are stored by column, the transpose of the matrix is what actually appears on the printed or plotted output. Blocks used for the first NL columns may now be re-used for subsequent groups of NL columns. This process is continued until all columns of the matrix have been processed. As many as five matrices may be handled during a single call to SEEMAT.

### 4.74.8 Subroutines

The plotter environment subroutines are utilized by SEEMAT. See section 3.4 for descriptions of the plotter utility routines.

#### 4.74.8.1 Subroutine MAPSET

1. Entry Points: MAPSET, MAP
2. Purpose: Converts physical units to plotter units for module SEEMAT.
3. Calling Sequence: CALL MAPSET (X1,Y1,X2,Y2,KI1,KJ1,KI2,KJ2,L)

CALL MAP (X,Y,KI,KJ)

X,Y,Xi,Yi = Physical coordinates

KI,KJ,KIi,KJi = Plotter coordinates

L = Output Format Flag, input

1 = KI,KJ are integer

2 = KI,KJ are real

The meaning of i follows:

i = 1 point is lower left corner of frame

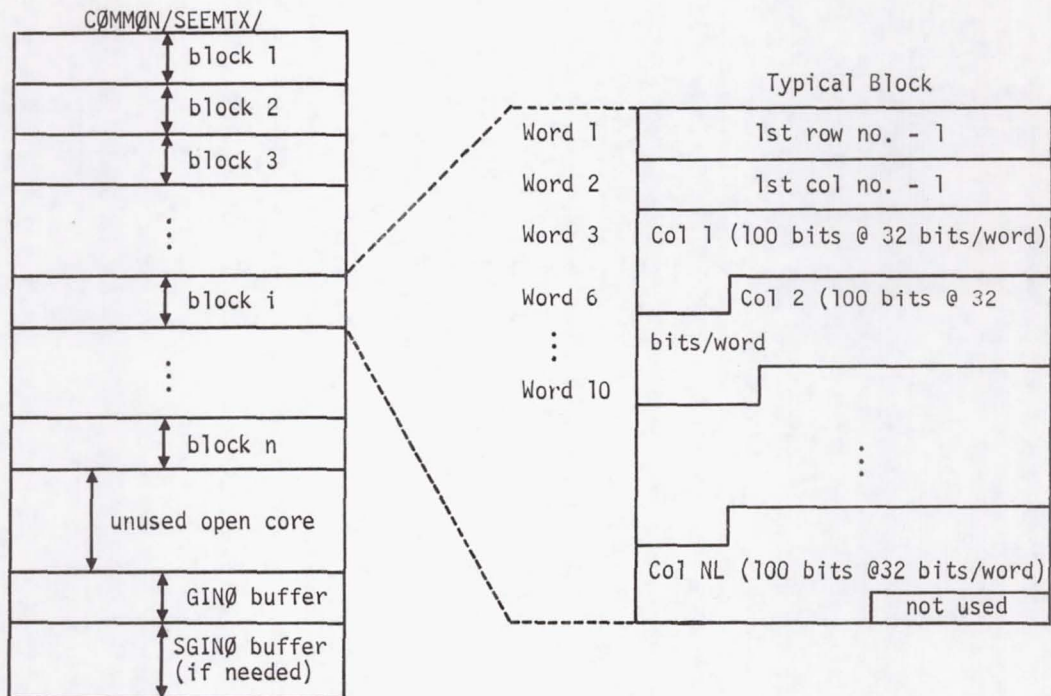
i = 2 point is upper right corner of frame

i = blank point is an arbitrary point on frame.

## MODULE FUNCTIONAL DESCRIPTIONS

### 4.74.9 Design Requirements

#### 4.74.9.1 Open Core Design





## OUTPUT MODULE SEEMAT (PICTORIAL MATRIX PRINTER)

### 4.74.9.2 Data Requirements and Restrictions

1. All nonpurged input data blocks must be matrices. Error diagnostics will occur in the unpacking routines if an attempt is made to input a table data block to SEEMAT.
2. If the number of blocks needed overflows the available open core (e.g., a large full matrix can do this), a nonfatal diagnostic message will be output on the System Output File and processing for that matrix will be terminated. The user may decrease NL by adding a Case Control Card LINE = NL as a means of overcoming this restriction (printer only). Since the type of matrix for which one is interested in seeing the topology is usually sparse and at least partially banded, this restriction should not prove serious.

### 4.74.10 Diagnostic Messages

Diagnostic conditions detected by SEEMAT are nonfatal and result in appropriate error messages and termination of the processing of the current input matrix data block. The one exception is the condition of no open core for GINØ buffers, which should not occur in practice.

## OUTPUT MODULE TABPT (TABLE PRINTER)

### 4.75 OUTPUT MODULE TABPT (TABLE PRINTER)

#### 4.75.1 Entry Point: TABPT.

#### 4.75.2 Purpose

To print table data blocks.

#### 4.75.3 DMAP Calling Sequence

TABPT TAB1,TAB2,TAB3,TAB4,TAB5// \$

#### 4.75.4 Input Data Blocks

TAB1 - Any NASTRAN data block.

TAB2 - Any NASTRAN data block.

TAB3 - Any NASTRAN data block.

TAB4 - Any NASTRAN data block.

TAB5 - Any NASTRAN data block.

Note: Any or all input data blocks can be purged.

#### 4.75.5 Output

Each word in a data block is identified as real, BCD, or integer and printed on the system output file. The trailer data is also printed.

#### 4.75.6 Parameters

None.

#### 4.75.7 Method

Subroutine TABPRT is called for each non-purged input file.

#### 4.75.8 Subroutines

TABPRT - See subroutine description, section 3.4.29.

## OUTPUT MODULE PRTMSG (MESSAGE WRITER)

### 4.76 OUTPUT MODULE PRTMSG (MESSAGE WRITER)

#### 4.76.1 Entry Point: PRTMSG

#### 4.76.2 Purpose

To process a data block of user-oriented messages.

#### 4.76.3 DMAP Calling Sequence

PRTMSG    MSG//    \$

#### 4.76.4 Input Data Blocks

MSG - Messages to be printed (if purged, nothing is done).

#### 4.76.5 Output Data Blocks

None

#### 4.76.6 Parameters

None

#### 4.76.7 Method

In addition to messages, the MSG data block may contain titles and subtitles. Before the first message is printed, a new page is started. From then on, a message count is maintained so as to start another new page when the maximum number of lines per page is exceeded. All messages are assumed to be only one line long. However, there is logic included to provide for messages of more than one line, forcing a new page at any time, and the alteration of titles and subtitles at any time. The description below of subroutine WRTMSG details all the included logic capability.

#### 4.76.8 Subroutines

PRTMSG uses the utility routines EJECT and FREAD (see sections 3.4.62 and 3.4.15).

##### 4.76.8.1 Subroutine Name: PRTMSG



## MODULE FUNCTIONAL DESCRIPTIONS

1. Entry Point: PRTMSG

2. Purpose: To print the user messages in the MSG data block.

3. Calling Sequence: CALL PRTMSG

COMMON/OUTPUT/TITLE(32,6) - See OUTPUT miscellaneous table description in section 2.5.

Where:

TITLE = NASTRAN title, subtitle, label, and three extra subtitles.

4. Method: Open the MSG data block, and skip record 0. If the MSG data block does not exist, nothing else is attempted. Otherwise, the three extra subtitles are set to all blanks, and WRTMSG is called.

### 4.76.8.2 Subroutine Name: WRTMSG (General purpose subroutine)

1. Entry Point: WRTMSG

2. Purpose: To process a data block of user-oriented messages.

3. Calling Sequence: CALL WRTMSG (MSG)

COMMON/SYSTEM/ - See SYSTEM table description in section 2.4.1.8.

Where:

MSG = GINØ file name of the MSG data block.

and in /SYSTEM/

MAXLIN = Maximum number of lines permitted per page.

CØUNT = Number of lines thus far printed on the current page.

4. Method:

a. Save the current NASTRAN title, subtitle, and label. Force the first message to start on a new page (CØUNT=MAXLIN).

b. Read one word (N) from MSG. If an end-of-record condition occurs, force the first message in the next record to start on a new page (CØUNT=MAXLIN). Then repeat this step.

c. If  $N < 0$ , the next 32 words are assumed to be a replacement for TITLE (1-32,N). Force the next message to start on a new page (CØUNT=MAXLIN). Repeat step b.

## OUTPUT MODULE PRMSG (MESSAGE WRITER)

- d. If  $N > 0$ , a list and format follow. The next  $N$  items are assumed to be the list items. If  $N = 0$ , only a format follows.
- e. Read one word (NF) from MSG. If  $NF < 0$ ,  $NF$  = number of lines to be generated by this message (repeat this step). If  $NF = 0$ , this message will be printed starting on a new page. (COUNT=MAXLIN, repeat this step). Unless otherwise instructed, this subroutine assumes that each message will generate only 1 line of output. In either case, integer function EJECT is called to maintain the page line count. If this message will not fit on this page, any extra title(s) explicitly specified are printed below the NASTRAN title, subtitle, and label.
- f. If  $NF > 0$ ,  $NF$  = size of the format (the format must be a continuous string of characters, contrary to the usual NASTRAN method of specifying at most 4 characters per word). The next  $NF$  words are assumed to be the format to be used with the list items read in step d, in one the following FORTRAN statements:

or           WRITE (MØ,FØR) [if no list is specified].  
              WRITE (MØ,FØR) (LIST(I), I = 1,N)

Repeat step b.

- g. When the end of the MSG data block is encountered in step b, the NASTRAN title, subtitle, and label are restored, and the MSG data block is closed with a rewind.
5. Design Requirements:
- a. The message data block (MSG) must be opened before calling WRTMSG.
  - b. In general, a set of messages is one record of the data block. Each set of messages will start on a new page.

### 4.76.9 Design Requirements

Open core is defined at /XXPMSG/, and is used only for one GINØ buffer which is defined at the beginning of open core.



## OUTPUT MODULE PRTPARM (PARAMETER AND DMAP MESSAGE PRINTER)

### 4.77 OUTPUT MODULE PRTPARM (PARAMETER AND DMAP MESSAGE PRINTER)

#### 4.77.1 Entry Point: PRTPRM

#### 4.77.2 Purpose

To print parameter values and DMAP messages.

#### 4.77.3 DMAP Calling Sequence

PRTPARM //C,N, $\alpha$ /C,N, $\beta$  \$

#### 4.77.4 Input Data Blocks

None

#### 4.77.5 Output Data Blocks

None

#### 4.77.6 Parameters

- $\alpha$  - Integer value (no default value)
- $\beta$  - BCD value (default value = XXXXXXXX)

#### 4.77.7 Method

As a parameter printer, use  $\alpha = 0$ . There are two options:

1.  $\beta$  = parameter name will cause the printout of the value of that parameter.

Example: PRTPARM //C,N,0/C,N,LUSET \$

2.  $\beta$  = XXXXXXXX will cause the printout of the values of all parameters in the current XVPS. Since this is the default value, it need not be specified.

Example: PRTPARM //C,N,0 \$

As a DMAP message printer, use  $\alpha \neq 0$ . There are two options:

1.  $\alpha > 0$  causes the printout of the  $j^{\text{th}}$  message of category  $\beta$  where  $j = |\alpha|$  and  $\beta$  is one of the values shown below. (The number of messages available in each category is also shown).

Example: PRTPARM //C,N,1/C,N,DMAP \$



## MODULE FUNCTIONAL DESCRIPTIONS

2.  $\alpha < 0$  causes the same action as  $\alpha > 0$  with the additional action of program termination. Thus, PRTPARM may be used as a fatal message printer.

Example: PRTPARM //C,N,-2/C,N,PLA \$

### 4.77.8 Remarks

1.  $\beta$  is always a value.

2. TABLE OF  $\beta$  CATEGORY VALUES

| <u>Rigid Format</u>                         | <u>Value of Beta</u> | <u>Number of Messages</u> |
|---|----------------------|---------------------------|
| Static Analysis                             | STATICS              | 4                         |
| Static Analysis with Inertia Relief         | INERTIA              | 4                         |
| Normal Mode Analysis                        | MØDES                | 4                         |
| Static Analysis with Differential Stiffness | DIFFSTIF             | 6                         |
| Buckling Analysis                           | BUCKLING             | 7                         |
| Piecewise Linear Analysis                   | PLA                  | 6                         |
| Direct Complex Eigenvalue Analysis          | DIRCEAD              | 5                         |
| Direct Frequency and Random Response        | DIRFRRD              | 6                         |
| Direct Transient Response                   | DIRTRD               | 5                         |
| Modal Complex Eigenvalue Analysis           | MDLCEAD              | 6                         |
| Modal Frequency and Random Response         | MDLFRRD              | 7                         |
| Modal Transient Response                    | MDLTRD               | 7                         |
| DMAP  | DMAP                 | 1                         |

3. For details on error messages for the  $i^{\text{th}}$  Rigid Format see section 3.(i + 1).2 in the User's Manual.

### 4.77.9 Subroutines

PRTPRM has no auxiliary subroutines.

OUTPUT MODULE PRTPARM (PARAMETER AND DMAP MESSAGE PRINTER)

4.77.10 Diagnostic Messages

Values of  $\alpha$  and  $\beta$  inconsistent with the above under "Method" will result in diagnostic messages from PRTPARM.

## MATRIX MODULE ADD (ADD TWO MATRICES)

### 4.78 MATRIX MODULE ADD (ADD TWO MATRICES)

#### 4.78.1 Entry Point: DADD

#### 4.78.2 Purpose

To compute  $[C] = \alpha[A] + \beta[B]$ .

#### 4.78.3 DMAP Calling Sequence

ADD A,B/C/C,Y,ALPHA=(1.0,2.0)/C,Y,BETA=(3.0,4.0) \$

#### 4.78.4 Input Data Blocks

A - Any matrix  $\neq$  B

B - Any matrix  $\neq$  A

Note: A and/or B can be purged.

#### 4.78.5 Output Data Blocks

C - Matrix.

The type of C is maximum of the types of A, B,  $\alpha$ ,  $\beta$ . The shape of C is the shape of A if A is present. Otherwise it is that of B.

Note: C cannot be purged.

#### 4.78.6 Parameters

ALPHA - Input-complex-no default value. This is the scalar multiplier for A.

BETA - Input-complex-no default value. This is the scalar multiplier for B.

Note: If  $\text{Im}(\alpha)$  or  $\text{Im}(\beta) = 0.0$ , the parameter will be considered real.

#### 4.78.7 Method

If [A] is not purged, the number of columns, rows, and form of [C] = number of columns, rows, and form of [A]. Otherwise the [B] descriptors are used. The type of [C] is the maximum compatible type of [A], [B], ALPHA and BETA. ALPHA and BETA are assumed to be real if their imaginary parts are zero.



## MODULE FUNCTIONAL DESCRIPTIONS

### 4.78.8 Subroutines

ADD - See subroutine description, Section 3.5.10.

### 4.78.9 Design Requirements

Open core is defined at /DADDA/.

### 4.78.10 Diagnostic Messages

None.

## MATRIX MODULE MPYAD (MULTIPLY ADD)

### 4.79 MATRIX MODULE MPYAD (MULTIPLY ADD)

#### 4.79.1 Entry Point: DMPYAD

#### 4.79.2 Purpose

MPYAD performs the multiplication of two matrices and, optionally, addition of a third matrix to the product:  $[D] = [A][B] \pm [C]$

#### 4.79.3 DMAP Calling Sequence

MPYAD     A,B,C/D/C,N,T/C,N,SIGNAB/C,N,SIGNC/C,N,PREC     \$

#### 4.79.4 Input Data Blocks

- A - Left hand matrix in the matrix product  $[A][B]$
- B - Right hand matrix in the matrix product  $[A][B]$
- C - Matrix to be added to  $[A][B]$

#### Notes:

1. If no matrix is to added, C must be purged.
2. A, B, C must be physically different data blocks.
3. A and B must not be purged.
4. Either A or B (but not both) may be a NASTRAN diagonal matrix. In this case, C must be purged.

#### 4.79.5 Output Data Block

- D - Matrix resulting from the MPYAD operation.

Note: D may not be purged.

#### 4.79.6 Parameters

- T     - Integer-input-no default.     T     =  $\begin{cases} 1, & \text{perform } [A]^T[B] \\ 0, & \text{perform } [A][B] \end{cases}$

## MODULE FUNCTIONAL DESCRIPTIONS

|                                    |  |
|------------------------------------|--|
| SIGNAB - Integer-input-no default. | $\text{SIGNAB} = \begin{cases} +1, & \text{perform } [A][B] \\ -1, & \text{perform } -[A][B] \end{cases}$  |
| SIGNC - Integer-input-no default.  | $\text{SIGNC} = \begin{cases} +1, & \text{add } [C] \\ -1, & \text{subtract } [C] \end{cases}$   |
| PREC - Integer-input-no default.   | $\text{PREC} = \begin{cases} 1, & \text{elements of } [D] \text{ will be output in} \\ & \text{single-precision.} \\ 2, & \text{elements of } [D] \text{ will be output in} \\ & \text{double-precision.} \end{cases}$ |

### 4.79.7 Examples

1.  $[D] = [A][B] + [C]$  (D double precision)  
MPYAD A,B,C/D/C,N,0/C,N,1/C,N,2 \$
2.  $[D] = [A]^T[B] - [C]$  (D single precision)  
MPYAD A,B,C/D/C,N,1/C,N,1/C,N,-1/C,N,1 \$
3.  $[D] = -[A][B]$  (D double precision)  
MPYAD A,B,/D/C,N,0/C,N,-1/C,N,0/C,N,2 \$

### 4.79.8 Method

DMPYAD reads the trailers for the data blocks A, B and C. /MPYADX/ is initialized. If neither [A] nor [B] is diagonal, MPYAD is called, the trailer for D is written, and the module exits. Otherwise, /DMPYX/ is initialized, and DMPY is called to perform the diagonal multiplication. If the matrix [C] is present, /ADDX/ is initialized, and ADD is called to perform the matrix addition. The trailer for D is written and the module exits.

### 4.79.9 Subroutines

DMPYAD calls the following matrix operations:

- MPYAD (see section 3.5.12 for details)
- DMPY (see section 3.5.21 for details)
- ADD (see section 3.6.10 for details)

### 4.79.10 Design Requirements

#### 4.79.10.1 Allocation of core storage



## MATRIX MODULE MPYAD (MULTIPLY ADD)

See descriptions for MPYAD, DMPY and ADD.

### 4.79.10.2 Environment

The module MPYAD consists of one subroutine, DMPYAD. One scratch file is used. Three common blocks define open core, one for each of the three overlay segments containing the matrix operations:

/MPYA1D/ included at end of segment containing MPYAD.

/MPYA2D/ included at end of segment containing DMPY.

/MPYA3D/ included at end of segment containing ADD.

MATRIX MODULE SØLVE (SOLVES THE MATRIX EQUATION  $[A][X] = [B]$ )

#### 4.80 MATRIX MODULE SØLVE (SOLVES THE MATRIX EQUATION $[A][X] = [B]$ )

##### 4.80.1 Entry Point: SØLVE

##### 4.80.2 Purpose

To solve the matrix equation,

$$[A][X] = \pm [B] \quad (1)$$

##### 4.80.3 DMAP Calling Sequence

SØLVE    A,B/X/V,Y,SYM/V,Y,SIGN/V,Y,PREC/V,Y,TYPE    \$

##### 4.80.4 Input Data Blocks

A                - A square real or complex matrix.

B                - A rectangular matrix.

Note: If B is purged, the identity matrix is assumed.

##### 4.80.5 Output Data Blocks

X                - A rectangular matrix.

##### 4.80.6 Parameters

SYM - Input-integer-default = 0    { 0 - use unsymmetric decomposition  
  1 - use symmetric decomposition.  
  -1 - try symmetric decomposition, use  
  unsymmetric if [A] is singular.

SIGN - input-integer-default = 1    { 1 - solve  $[A][X] = [B]$   
  -1 - solve  $[A][X] = -[B]$

PREC - Input-integer-default = 1    { 1 - use single precision arithmetic  
  2 - use double precision arithmetic

TYPE - Input-integer-default = 1    { 1 - output type of matrix [X] is real  
  single precision  
  2 - output type of matrix [X] is real  
  double precision

## MODULE FUNCTIONAL DESCRIPTIONS

- 3 - output type of matrix [X] is complex  
single precision
- 4 - output type of matrix [X] is complex  
double precision

### 4.80.7 Method

Depending upon the SYM flag and the type of [A], either SDCOMP CDCOMP, or DECOMP is called to form

$$[A] = [L][U] . \quad (2)$$

FBS or GFBS is called to solve

$$[L][Y] = \pm [B] , \quad (3)$$

and

$$[U][X] = [Y] . \quad (4)$$

### 4.80.8 Subroutines

The above mentioned subroutines are the only ones called by SOLVE and are documented in section 3.5.

### 4.80.9 Design Requirements

The appropriate subroutines should be referenced for the design requirements peculiar to each routine.

### 4.80.10 Diagnostic Messages

The individual routines should be referred to for diagnostic messages.



## MATRIX MODULE DECØMP (MATRIX DECOMPOSITION)

### 4.81 MATRIX MODULE DECØMP (MATRIX DECOMPOSITION)

#### 4.81.1 Entry Point: DDCØMP

#### 4.81.2 Purpose

To decompose a square matrix [A] into lower [L] and upper [U] triangular factors.

#### 4.81.3 DMAP Calling Sequence

DECØMP    A/L,U/V,Y,KSVM/V,Y,CHØLSKY/V,N,MINDIAG/V,N,DET/V,N,PØWER/V,N,SING    \$

#### 4.81.4 Input Data Blocks

A                -    A square matrix.

#### 4.81.5 Output Data Blocks

L                -    Lower triangular factor of [A].

U                -    Non-standard upper triangular factor of [A].

#### 4.81.6 Parameters

KSVM            -    Input-integer-no default. 1, use symmetric decomposition. 0, use unsymmetric decomposition.

CHØLSKY        -    Input-integer-default = 0. 1, use Cholesky decomposition. 0, do not use Cholesky decomposition.

MINDIAG        -    Output-real-no default. The minimum diagonal term of [U].

DET             -    Output-complex single precision-no default. The scaled value of the determinant of [A].

PØWER           -    Output-integer-no default. Integer PØWER of 10 by which DET should be multiplied to obtain the determinant of [A].

SING            -    Output-integer-no default. SING is set to -1 if [A] is singular.

#### 4.81.7 Method

Depending upon the type of [A] and the KSVM flag, a calling sequence is set up, and either

## MODULE FUNCTIONAL DESCRIPTIONS

CDCØMP, DECØMP, or SDCØMP is called.

### 4.81.8 Subroutines

The major subroutines used are DECØMP, CDCØMP and SDCØMP. Descriptions of these subroutines can be found in sections 3.5.15, 3.5.16, and 3.5.14 respectively.

### 4.81.9 Design Requirements

The individual subroutine writeups should be consulted for the particular restrictions of each routine.

### 4.81.10 Diagnostic Messages

See the appropriate subroutine descriptions.

## MATRIX MODULE FBS (FORWARD-BACKWARD SUBSTITUTION)

### 4.82 MATRIX MODULE FBS (FORWARD-BACKWARD SUBSTITUTION)

#### 4.82.1 Entry Point: DFBS

#### 4.82.2 Purpose

To solve the equation,

$$[L] [U] [X] = \pm [B] \quad (1)$$

where [L] and [U] are the upper and lower triangular factors obtained via matrix module DECØMP.

#### 4.82.3 DMAP Calling Sequence

FBS    L,U,B/X/C,N,A/C,N,B/C,N,C/C,N,D    \$

#### 4.82.4 Input Data Blocks

- L,U            - Matrices output from module DECØMP.
- B              - Rectangular matrix.

#### 4.82.5 Output Data Blocks

- X              - Rectangular matrix.

#### 4.82.6 Parameters

- A            - Input-integer-no default.  $\left\{ \begin{array}{l} 1 \text{ matrix } [L][U] \text{ is symmetric} \\ 0 \text{ matrix } [L][U] \text{ is unsymmetric} \end{array} \right.$
- B            - Input-integer-no default.  $\left\{ \begin{array}{l} 1 \text{ solve } [L][U][X] = [B] \\ -1 \text{ solve } [L][U][X] = -[B] \end{array} \right.$
- C            - Input-integer-no default.  $\left\{ \begin{array}{l} 1 - \text{ use single precision arithmetic} \\ 2 - \text{ use double precision arithmetic} \end{array} \right.$
- D            - Input-integer-no default.  $\left\{ \begin{array}{l} 1 - \text{ output } [X] \text{ in single precision} \\ 2 - \text{ output } [X] \text{ in double precision} \end{array} \right.$

#### 4.82.7 Method

Depending upon the value of the parameter A, either FBS or GFBS is called.



## MODULE FUNCTIONAL DESCRIPTIONS

### 4.82.8 Subroutines

The above routines are the only ones called by DFBS. Their descriptions are given in section 3.5.17 for FBS and 3.5.19 for GFBS.

### 4.82.9 Design Requirements

The appropriate routines should be referenced for their individual requirements.

### 4.82.10 Diagnostic Messages

The individual subroutines should be referred to for the messages.

## MATRIX MODULE PARTN (PARTITION A MATRIX)

### 4.83 MATRIX MODULE PARTN (PARTITION A MATRIX)

#### 4.83.1 Entry Point: PARTN1

#### 4.83.2 Purpose

To partition [A] into [A11], [A12], [A21] and [A22]:

$$[A] \Rightarrow \begin{bmatrix} A11 & | & A21 \\ \hline A12 & | & A22 \end{bmatrix}. \quad (1)$$

#### 4.83.3 DMAP Calling Sequence

PARTN A,RP,CP/A11,A12,A21,A22/V,Y,SYM/V,Y,TYPE/V,Y,FØRM1/V,Y,FØRM2/V,Y,FØRM3/V,Y,FØRM4 \$

#### 4.83.4 Input Data Blocks

A - Matrix to be partitioned.

RP - Row partitioning vector - single precision column vector.

CP - Column partitioning vector - single precision column vector.

#### Notes:

1. If A is purged, PARTN returns.
2. If RP is purged, A is partitioned as follows:

$$[A] \Rightarrow \begin{bmatrix} A11 \\ \hline A12 \end{bmatrix}. \quad (2)$$

3. If CP is purged and  $SYM > 0$ , A is partitioned as follows:

$$[A] \Rightarrow [A11 | A21]. \quad (3)$$

4. If CP is purged and  $SYM \leq 0$ , A is partitioned as follows:

$$[A] \Rightarrow \begin{bmatrix} A11 & | & A21 \\ \hline A12 & | & A22 \end{bmatrix}, \quad (4)$$

where RP is used as both the row and column partitioner.

5. RP and CP cannot both be purged.

## MODULE FUNCTIONAL DESCRIPTIONS

### 4.83.5 Output Data Blocks

A11 - Partition of A.

A12 - Partition of A.

A21 - Partition of A.

A22 - Partition of A.

#### Notes:

1. Any or all output data blocks can be purged.
2. For the shape of outputs (number of rows and columns) see Section 4.83.7 below.

### 4.83.6 Parameters

|       |   |  |
|-------|---|--|
| SYM   | - Input-integer-default=-1.   | SYM chooses between a symmetric partition and an unsymmetric partition. If $SYM \leq 0$ , CP is used as RP. If $SYM > 0$ , CP and RP are distinct.   |
| TYPE  | $\left\{ \begin{array}{l} \text{Input-integer-default} = 0. \\ \text{Output-integer} \end{array} \right.$ | $\left\{ \begin{array}{l} \text{Type of output matrices } 0 \leq TYPE \leq 4; \text{ default is logical choice based on matrix A and requested partitions.} \\ \text{Type of output matrices produced.} \end{array} \right.$ |
| FØRM1 | $\left\{ \begin{array}{l} \text{Input-integer-default} = 0. \\ \text{Output-integer} \end{array} \right.$ | $\left\{ \begin{array}{l} \text{Form of A11, } 0 < FØRM1 \leq 8; \text{ default uses logical choice based on matrix A.} \\ \text{Form of A11 produced.} \end{array} \right.$   |
| FØRM2 | -   | Same as FØRM1 but applied to A12.  |
| FØRM3 | -   | Same as FØRM1 but applied to A21.  |
| FØRM4 | -   | Same as FØRM1 but applied to A22.  |

### 4.83.7 Method

Let N1 = number of non-zero terms in RP.

Let N2 = number of non-zero terms in CP.

Let NRØWA = number of rows in A.

Let NCØLA = number of columns in A.

CASE:1 RP purged.

A11 is a  $(NRØWA-N2) \times NCØLA$  matrix.

A12 is a  $N2 \times NCØLA$  matrix.

A21 is not written.

A22 is not written.



## MATRIX MODULE PARTN (PARTITION A MATRIX)

CASE 2: CP purged and  $\text{SYM} > 0$ .

A11 is a  $\text{NRØWA} \times (\text{NCØLA} - \text{N1})$  matrix.

A12 is not written.

A21 is a  $\text{NRØWA} \times \text{N1}$  matrix.

A22 is not written.

CASE 3: CP purged and  $\text{SYM} \leq 0$ .

A11 is a  $(\text{NRØWA} - \text{N1}) \times (\text{NCØLA} - \text{N1})$  matrix.

A12 is a  $\text{N1} \times (\text{NCØLA} - \text{N1})$  matrix.

A21 is a  $(\text{NRØWA} - \text{N1}) \times \text{N1}$  matrix.

A22 is a  $\text{N1} \times \text{N1}$  matrix.

CASE 4: Neither RP nor CP purged.

A11 is a  $(\text{NRØWA} - \text{N2}) \times (\text{NCØLA} - \text{N1})$  matrix.

A12 is a  $\text{N2} \times (\text{NCØLA} - \text{N1})$  matrix.

A21 is a  $(\text{NRØWA} - \text{N2}) \times \text{N1}$  matrix.

A22 is a  $\text{N2} \times \text{N1}$  matrix.

In general if  $a_{ij} \in [A]$ , then:

$$a_{ij} \in [A11] \text{ if } \text{RP}(J) = \text{CP}(I) = 0$$

$$a_{ij} \in [A12] \text{ if } \text{CP}(I) \neq 0, \text{RP}(J) = 0$$

$$a_{ij} \in [A21] \text{ if } \text{CP}(I) = 0, \text{RP}(J) \neq 0$$

$$a_{ij} \in [A22] \text{ if } \text{CP}(I) \neq 0, \text{RP}(J) \neq 0$$

### 4.83.8 Subroutines

#### 4.83.8.1 Subroutine Name: PARTN2

1. Entry Point: PARTN2

2. Purpose: Initialization routine for PARTN1 and MERGE1. It calls PARTN3 to build the bit strings from the partitioning vectors CP and RP and sets default options based on SYM.

## MODULE FUNCTIONAL DESCRIPTIONS

### 3. Calling Sequence: CALL PARTN2 (CP,RP,CØRE,BUF)

CP = GINØ file name of column partitioning vector  
RP = GINØ file name of row partitioning vector  
CØRE = Location of first word open core  
BUF = Location of GINØ buffer.

#### 4.83.8.2 Subroutine Name: PARTN3

##### 1. Entry Point: PARTN3

##### 2. Purpose: Builds bit strings as directed by PARTN2.

##### 3. Calling Sequence: CALL PARTN3 (FILE,SIZE,ØNES,IZ,NZ,HERE,BUF,CØRE)

FILE = GINØ file name  
SIZE = Length of partitioning vector  
ØNES = Number of non-zero terms in vector  
IZ = Pointer to working core  
NZ = Length of working core  
HERE = Logical Flag  
BUF = Location of GINØ buffer  
CØRE = Location of open core

#### 4.83.9 Design Requirements

Open core is defined at /PARTN1/.

#### 4.83.10 Diagnostic Messages

Messages 2166, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 3002, and 3008 may be issued.

## MATRIX MODULE MERGE (MERGE MATRICES TOGETHER)

### 4.84 MATRIX MODULE MERGE (MERGE MATRICES TOGETHER)

#### 4.84.1 Entry Point: MERGE1

#### 4.84.2 Purpose

To form

$$[A] \Leftarrow \left[ \begin{array}{c|c} A11 & A21 \\ \hline A12 & A22 \end{array} \right]. \quad (1)$$

#### 4.84.3 DMAP Calling Sequence

MERGE A11,A12,A21,A22,RP,CP/A/V,Y,SYM/V,Y,TYPE/V,Y,FØRM \$

#### 4.84.4 Input Data Blocks

A11 - Matrix ≠ A12, A21, A22.

A12 - Matrix ≠ A11, A21, A22.

A21 - Matrix ≠ A11, A12, A22.

A22 - Matrix ≠ A11, A12, A21.

RP - Row partitioning vector - single precision vector.

CP - Column partitioning vector - single precision vector.

#### Notes:

1. Any or all of A11, A12, A21, A22 can be purged which implies  $[AIJ] = [0]$ .
2. RP and CP cannot both be purged.
3. See method section for meaning when RP or CP is purged.

#### 4.84.5 Output Data Blocks

A - Merged matrix from A11, A12, A21, A22.

Notes: A cannot be purged.



## MODULE FUNCTIONAL DESCRIPTIONS

### 4.84.6 Parameters

- SYM - Input-integer-no default.  $\text{SYM} \leq 0$ , CP is used as RP.  $\text{SYM} > 0$ , CP and RP are distinct.
- TYPE - Input-integer-no default. Type of A. 1 implies A is real single precision, 2 implies A is real double precision, 3 implies A is complex single precision, 4 implies A is complex double precision.
- FØRM - Input-integer-no default. Form of A (see section 2.2).

### 4.84.7 Method

MERGE is the inverse of PARTN in the sense that if A11, A12, A21, A22 were produced by PARTN using RP, CP, FØRM, SYM, and TYPE from A, MERGE will reproduce A. See PARTN (section 4.83) for options on RP, CP and SYM.

### 4.84.8 Subroutines

Subroutines PARTN2 and PARTN3 are used. These routines are described in Section 4.83.

### 4.84.9 Design Requirements

Open core is defined at /MERGE1/.

### 4.84.10 Diagnostic Messages

Messages 2161, 2162, 2163, 2164, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 3002, and 3008 may be issued.

## MATRIX MODULE TRNSP (TRANPOSE A MATRIX)

### 4.85 MATRIX MODULE TRNSP (TRANPOSE A MATRIX).

#### 4.85.1 Entry Point: DTRANP

#### 4.85.2 Purpose

To form  $[A]^T$  given  $[A]$ .

#### 4.85.3 DMAP Calling Sequence

TRNSP    A/AT    \$

#### 4.85.4 Input Data Blocks

A - Any matrix data block.

Note: If  $[A]$  is purged, TRNSP returns.

#### 4.85.5 Output Data Blocks

AT - The matrix transpose of  $[A]$ .

Note: AT cannot be purged.

#### 4.85.6 Parameters

None.

#### 4.85.7 Method

Subroutine TRNSP is called.

#### 4.85.8 Subroutines

TRNSP - See subroutine description, section 3.5.25.

#### 4.85.9 Design Requirements

Open core is defined at /DTRANX/. Eight scratch files are used.

## MATRIX MODULE SMPYAD (STRING MULTIPLY ADD)

### 4.86 MATRIX MODULE SMPYAD (STRING MULTIPLY ADD)

#### 4.86.1 Entry Point: SMPYAD

#### 4.86.2 Purpose

To multiply a series of matrices together.

#### 4.86.3 DMAP Calling Sequence

SMPYAD A,B,C,D,E,F/G/C,N,N/C,N,SIGNX/C,N,SIGNF/C,N,PG/C,N,TA/C,N,TB/C,N,TC/C,N,TD \$

#### 4.86.4 Input Data Blocks

$\left. \begin{matrix} A \\ B \\ C \\ D \\ E \end{matrix} \right\}$  - Up to 5 matrices to be multiplied together, from left to right.

F - Matrix to be added to the above product.

#### Notes:

1. If one of the five multiplication matrices is required in the product (see parameter "N" below) and is purged, the multiplication will not be done.
2. If the F matrix is purged, no matrix will be added to the product.

#### 4.86.5 Output Data Blocks

G - Resultant matrix (may not be pre-purged).

#### 4.86.6 Parameters

- N - Number of matrices involved in the product - integer - input.
- SIGNX - Sign of the product matrix (e.g., [A][B][C][D][E]) - integer - input.  
1 for plus, -1 for minus.
- SIGNF - Sign of the matrix to be added to the product matrix - integer - input.  
1 for plus, -1 for minus.
- PG - Output precision of the final result - integer - input.  
1 for single precision, 2 for double precision.



## MODULE FUNCTIONAL DESCRIPTIONS

TA }  
TB } - Transpose indicators for the [A][B][C] and [D] matrices (1 if transposed  
TC } matrix to be used in the product; 0 if untransposed) - integer - input.  
TD }

Note: All the parameters except "N" have default values. They are these:

1. SIGNX = 1 (sign of product is plus)
2. SIGNF = 1 (sign of added matrix is plus)
3. PG = 1 (single precision result)
4. TA }  
TB } = 0 (use untransposed [A], [B], [C], and [D] matrices in the product)  
TC }  
TD }

### 4.86.7 Method

The method is the same as for the MPYAD module with one exception and one addition:

1. None of the matrices may be diagonal.
2. Except for the final product, all intermediate matrix products are generated in double precision.

The matrices are multiplied together from right-to-left, i.e., the first product calculated is the product of matrix n-1 and matrix n.

### 4.86.8 Subroutines

MPYAD is called (see section 3.5.12 for details).

### 4.86.9 Design Requirements

1. Two scratch files are required.
2. Open core is the /MPYADX/ common block, the same one as used by the MPYAD module, (see section 4.79).

## STRUCTURAL ELEMENT DESCRIPTIONS

### 4.87 STRUCTURAL ELEMENT DESCRIPTIONS

The finite structural element subroutines used in NASTRAN have a number of different calculations associated with them. These subroutines are found in the modules SMA1, SMA2, SSG1, SDR2, DSMG1, PLA3 and PLA4.

All modules excluding IFP having anything to do with the NASTRAN structural elements, their geometry, or associated data blocks, use the basic element data found in common block /GPTA1/. /GPTA1/ is set in its own block data subprogram, and/or by (in the presence of dummy-user-elements) the routine DELSET. Refer to Section 2.5.2.1 for further information regarding /GPTA1/.

The element subroutines in the SMA1 (Structural Matrix Assembler - Phase 1) module generate element stiffness matrix partitions. The stiffness matrix,  $[K]$ , for a structural element consists of a 6 by 6 partition for each combination of the connected grid points. For example, a RØD element is connected to two grid points, "a" and "b". The stiffness matrix partitions are:  $[K_{aa}]$ ,  $[K_{ab}]$ ,  $[K_{ba}]$  and  $[K_{bb}]$ . A triangular element (e.g., TRMEM) is connected to three points. It will generate nine partitions:  $[K_{aa}]$ ,  $[K_{ab}]$ ,  $[K_{ac}]$ ,  $[K_{ba}]$ ,  $[K_{bb}]$ ,  $[K_{bc}]$ ,  $[K_{ca}]$ ,  $[K_{cb}]$  and  $[K_{cc}]$ . In order to generate a particular partition,  $[K_{ij}]$ , it is often necessary to generate  $[K]$ . However, only those partitions  $[K_{ij}]$ , where  $i$  is the pivot point (see section 1.8) and  $j = 1, 2, \dots, n$  ( $n$  being the number of grid points associated with the element), are output by an element stiffness matrix generation subroutine, e.g., KRØD. These partitions are output from an element subroutine in the form of calls to the "insertion" subroutine SMA1B (see Section 4.27). There is one call for each 6 by 6 partition if the element is a structural element, and one call for each 1 by 1 "partition" if the element is a scalar element. The unused partitions are recalculated and used when  $j \neq i$  appears as a pivot point in a subsequent ECPT record. An alternate procedure for matrix generation, which is not used, would be to calculate all of the element matrices once and store them on an auxiliary storage unit for use when needed. The alternate procedure is less efficient for large problems, where efficiency really counts, because the recalculation time is less than the time required to recover element matrices from the auxiliary unit.

Element structural damping matrices,  $[K^4]$ , are proportional to the element stiffness matrices, the proportionality constant being  $g_e$ , the structural damping coefficient input on a material (e.g., MAT1) bulk data card. An element stiffness matrix generation routine, e.g., KRØD, of module SMA1 will output, through the calling sequence to subroutine SMA1B: 1) an element stiffness



## MODULE FUNCTIONAL DESCRIPTIONS

matrix partition, 2) the structural damping coefficient, and 3) a flag, which will signal SMA1B that the scalar multiplication of the matrix by the structural damping coefficient is to take place.

The element subroutines (e.g., MRØD, MCØNMx) in the SMA2 (Structural Matrix Assembler - Phase 2) module generate element mass matrix partitions. The remarks in the third paragraph above concerning element stiffness matrix partitions apply here also when the reader makes the substitutions: "mass" for "stiffness", "[M]" for "[K]", "MRØD" for "KRØD", and "SMA2B" for "SMA1B".

Only the element VISC and DAMPi generate viscous damping terms which contribute to the damping matrix,  $[B_{gg}]$ , and conversely, the only elements which contribute to  $[B_{gg}]$  are the VISC and DAMPi elements. These terms are calculated in module SMA2. The damping matrix partitions are passed to subroutine SMA2B in a fashion similar to that for mass matrix partitions.

Element static loading functions due to temperature and enforced deformations are generated in the SSG1 (Static Solution Generator - Phase 1) module, and the mathematical descriptions for these functions are given in this Section (4.87). (See the Module Functional Description for SSG1, Section 4.41, for the equations governing direct applied loads and gravity loads.) The output of an element routine are load vectors which are placed in the  $\{P_g\}$  load vector (see Section 4.41).

Element stresses and forces due to displacements are calculated in the SDR2 (Stress Data Recovery - Phase 2) module. These calculations are performed in two phases. Phase 1 generates element stress matrices for each element for which the user has requested element stress and/or force output. These element stress matrices are written on a scratch file for use in phase 2. In phase 2, the displacement vector for the current subcase is read into core, and, for each element for which stress and/or force output is requested, the corresponding element stress matrix is read and passed to the phase 2 element subroutine. The phase 2 element subroutine then calculates element stresses and forces. A list of the stresses and forces output in phase 2 for each element is given in Sections 2.3.51 and 2.3.52 respectively.

Differential stiffness matrix partitions are calculated for some elements. These are calculated in module DSMG1 (Differential Stiffness Matrix Generator - Phase 1) for large displacement analysis and buckling problems. The output of an element routine of the DSMG1 module are the 6 by 6 differential stiffness matrix partitions,  $[K_{ij}^d]$ , where  $i$  is the pivot point. The "insertion" subroutine for module DSMG1, similar to subroutine SMA1B of module SMA1, is DS1b.



## STRUCTURAL ELEMENT DESCRIPTIONS

Nonlinear, plastic effects in the structure may be determined by solving for the element stress and modifying the elastic properties of an element in an iterative loop. Element stresses are calculated in the PLA3 (Piecewise Linear Analysis - Phase 3) module, and element stiffness matrices with modified elastic properties are calculated in the PLA4 (Piecewise Linear Analysis - Phase 4). The outputs of an element subroutine of the PLA3 module are: 1) element stresses, which have the same formats as the element stresses output from a phase 2 element subroutine of module SDR2, and 2) updated incremental stress data in the ESTNLI data block, which are used as input to the PLA3 module in the next pass of the Piecewise Linear Analysis (PLA) Rigid Format DMAP loop. The outputs of an element subroutine of the PLA4 module are: 1) element stiffness matrix partitions (the remarks on element stiffness matrix partitions in the second paragraph apply here as well, except that the "insertion" subroutine is PLA4B) and 2) updated incremental stress data in the ECPTNLI data block, which are used as input to the PLA4 module in the next pass of the PLA Rigid Format DMAP loop.

The following data are needed to generate the element matrices in the above modules.

1. Element Connection and Properties Table (ECPT) Data.
2. Transformation matrices,  $[T_i]$ , from the global coordinate system to the basic coordinate system.
3. Material Property Data.
4. Element Deformation Data (used only in modules SSG1, SDR2 and DSGM1).
5. Grid Point Temperature Data (used only in modules SSG1, SDR2 and DSGM1).

The ECPT data are input to an element subroutine by a module driver from the ECPT data block or the EST (Element Summary Table) data block. The data in each of these data blocks are identical, from an individual element subroutine point of view. The ECPT data block is used in modules SMA1, SMA2 and DSGM1; the EST data block is used in modules SSG1 and SDR2. For the special case of Piecewise Linear Analysis, the ECPTNL (Element Connection and Properties Table for Nonlinear Elements) data block is used in module PLA4, and the ESTNL (Element Summary Table for Nonlinear Elements) data block is used in module PLA3. The ECPT and EST data blocks are generated in the Table Assembler (TA1) module (see Section 4.26) from the following data blocks: ECT (Element Connection Table, Section 2.3.4.1), EPT (Element Property Table, Section 2.3.2.5), BGPDT (Basic Grid Point Definition Table, Section 2.3.3.5), and GPTT (Grid Point Temperature Table, Section 2.3.7.2).

## MODULE FUNCTIONAL DESCRIPTIONS

The ECPT data for an element consist of four separate parts: 1) connection data 2) property data, 3) basic grid point definition data and 4) the element temperature for material properties. The connection data consists of data on a connection bulk data card (e.g., CRØD), except for the property identification number (the property identification number on the connection and property cards is used only to relate the two cards during the assembly of the ECPT and EST data blocks, and it does not appear in either the connection data or property data). Note also that grid point identification numbers have been converted to internal numbers, Scalar Index List (SIL) numbers, which correspond to degrees of freedom numbers. Property data consist of data on a property bulk data card (e.g., PRØD) with the above noted exception. Basic grid point definition data consist of, for each grid point connecting the element, 1) the identification number of the coordinate system in which displacements are defined at the grid point and 2) the coordinates of the grid point in the basic coordinate system. The element temperature for material properties is the average value given for each element in the GPTT data block. This temperature is placed in the ECPT/EST data in the Table Assembler module from the element temperatures in the GPTT data block and the set identification number,  $n$ , from the TEMPERATURE(MATERIAL) =  $n$  card in the User's Case Control Deck. Note that  $n$  is transmitted to the Table Assembler via the tenth word of /SYSTEM/ (see Section 2.4.1.8).

The transformation matrices,  $[T_i]$ , from the global coordinate system to the basic coordinate system, are supplied to an element subroutine by the utility routines TRANSD and TRANSS. These utility routines use the CSTM (Coordinate System Transformation Matrices, Section 2.3.3.4) data block in conjunction with the basic grid point definition data at a point  $i$  to compute  $[T_i]$ . Hence all modules which deal with element calculations require the CSTM data block as input. TRANSD returns, to an element subroutine, a double precision matrix,  $[T_i]$ , used by element routines in the following modules which use double precision arithmetic: SMA1, SMA2, DSGM1 and PLA4; TRANSS returns, to an element subroutine, a single precision matrix,  $[T_i]$ , used by element routines in the following modules which use single precision arithmetic: SSG1, SDR2 and PLA3.

Material property data are contained in the MPT (Material Properties Table, Section 2.3.2.6) and the DIT (Direct Input Tables, Section 2.3.2.7) data blocks. Both of these data blocks are output from the IFP (Input File Processor) Preface module. The utility routine MAT (see Section 3.4.36) fetches required material property data for element routines. These data are returned in single precision form.



## STRUCTURAL ELEMENT DESCRIPTIONS

Element deformation data are contained in the EDT (Element Deformation Table, Section 2.3.2.8) data block which is output by the IFP Preface Module. Element deformation data is admissible only for the RØD (including CØNRØD), TUBE and BAR elements.

Element temperature data are contained in the GPTT (Grid Point Temperature Table, Section 2.3.7.2), which is output by the GP3 (Geometry Processor - Phase 3) module. The temperature data contained in this data block are used for static loading functions due to temperature.

Table 1 on the following page gives reference to the Theoretical and User's Manuals where more information on the elements can be found.



# MODULE FUNCTIONAL DESCRIPTIONS

Table 1. Structural Element References.

| Bulk Data Connection<br>Card Mnemonic | Programmer's Manual<br>Reference | User's Manual<br>Reference | Theoretical Manual<br>Reference |
|---------------------------------------|----------------------------------|----------------------------|---------------------------------|
| CAXIF2                                | 4.87.15                          | 1.8                        | 17.1                            |
| CAXIF3                                | 4.87.15                          | 1.8                        | 17.1                            |
| CAXIF4                                | 4.87.15                          | 1.8                        | 17.1                            |
| CBAR                                  | 4.87.2                           | 1.3.2                      | 5.2, 7.2                        |
| CCONEAX                               | 4.87.9                           | 1.3.6                      | 5.9                             |
| CDAMP1                                | 4.87.7                           | 1.3.8                      | 5.6                             |
| CDAMP2                                | 4.87.7                           | 1.3.8                      | 5.6                             |
| CDAMP3                                | 4.87.7                           | 1.3.8                      | 5.6                             |
| CDAMP4                                | 4.87.7                           | 1.3.8                      | 5.6                             |
| CELAS1                                | 4.87.7                           | 1.3.8                      | 5.6                             |
| CELAS2                                | 4.87.7                           | 1.3.8                      | 5.6                             |
| CELAS3                                | 4.87.7                           | 1.3.8                      | 5.6                             |
| CELAS4                                | 4.87.7                           | 1.3.8                      | 5.6                             |
| CFLUID2                               | 4.87.15                          | 1.7                        | 16.1                            |
| CFLUID3                               | 4.87.15                          | 1.7                        | 16.1                            |
| CFLUID4                               | 4.87.15                          | 1.7                        | 16.1                            |
| CHBDY                                 | 4.87.17                          |                            | 8.2                             |
| CHEXA1                                | 4.87.17                          | 1.3.9                      | 5.12                            |
| CHEXA2                                | 4.87.17                          | 1.3.9                      | 5.12                            |
| CMASS1                                | 4.87.7                           | 1.3.8                      | 5.6                             |
| CMASS2                                | 4.87.7                           | 1.3.8                      | 5.6                             |
| CMASS3                                | 4.87.7                           | 1.3.8                      | 5.6                             |
| CMASS4                                | 4.87.7                           | 1.3.8                      | 5.6                             |
| CMFREE                                | 4.87.15                          | 1.7                        | 16.1                            |
| CØNM1                                 | 4.87.8                           | 1.2.3                      | 5.5                             |
| CØNM2                                 | 4.87.8                           | 1.2.3                      | 5.5                             |
| CØNRØD                                | 4.87.1                           | 1.3.3                      | 5.2, 7.2                        |
| CQDMEM                                | 4.87.4                           | 1.3.5                      | 5.8, 7.3                        |
| CQDMEM1                               | 4.87.19                          | 1.3.5                      | 5.8                             |
| CQDMEM2                               | 4.87.20                          | 1.3.5                      | 5.8                             |
| CQDPLT                                | 4.87.5                           | 1.3.5                      | 5.8                             |
| CQUAD1                                | 4.87.6                           | 1.3.5                      | 5.8, 7.3                        |
| CQUAD2                                | 4.87.6                           | 1.3.5                      | 5.8, 7.3                        |
| CRØD                                  | 4.87.1                           | 1.3.3                      | 5.2, 7.2                        |
| CSHEAR                                | 4.87.3                           | 1.3.4                      | 5.3                             |
| CSLØT3                                | 4.87.16                          | 1.8                        | 17.1                            |
| CSLØT4                                | 4.87.16                          | 1.8                        | 17.1                            |
| CTETRA                                | 4.87.17                          | 1.3.9                      | 5.12                            |
| CTØRDRG                               | 4.87.12                          | 1.3.7                      | 5.11                            |
| CTRAPRG                               | 4.87.11                          | 1.3.7                      | 5.10                            |
| CTRBSC                                | 4.87.5                           | 1.3.5                      | 5.8                             |
| CTRIA1                                | 4.87.6                           | 1.3.5                      | 5.8, 7.3                        |
| CTRIA2                                | 4.87.6                           | 1.3.5                      | 5.8, 7.3                        |
| CTRIARG                               | 4.87.10                          | 1.3.7                      | 5.8                             |
| CTRMEM                                | 4.87.4                           | 1.3.5                      | 5.8, 7.3                        |
| CTRPLT                                | 4.87.5                           | 1.3.5                      | 5.8                             |
| CTUBE                                 | 4.87.1                           | 1.3.3                      | 5.2, 7.2                        |
| CTWIST                                | 4.87.3                           | 1.3.4                      | 5.3                             |
| CVISC                                 | 4.87.13                          | 1.3.3                      | 5.2                             |
| CWEDGE                                | 4.87.17                          | 1.3.9                      | 5.12                            |

Note: The bulk data connection and property card descriptions in Section 2 of the NASTRAN User's Manual should also be consulted.

## STRUCTURAL ELEMENT DESCRIPTIONS

### 4.87.1 The RØD, CØNRØD and TUBE Elements

#### 4.87.1.1 Input Data for the RØD, TUBE, CØNRØD Elements

1. The ECPT/EST entries for the RØD and CØNRØD are:

| <u>Symbol</u>  | <u>Description</u>  |
|--|---|
| $SIL_a, SIL_b$   | Scalar indices for grid points a and b                              |
| $\left. \begin{matrix} N_a, X_a, Y_a, Z_a \\ N_b, X_b, Y_b, Z_b \end{matrix} \right\}$ | Local coordinate system number and basic coordinates of grid points |
| Mat I. D.  | Material identification number                                      |
| A  | Cross-section area  |
| J  | Polar inertia   |
| $\mu$  | Nonstructural mass per unit length                                  |
| C  | Shear stress coefficient  |
| $t_\mu$  | Temperature for material properties                                 |

2. The TUBE element has the same characteristics as the RØD except for different input properties. The TUBE has d, the outside diameter, and t, the thickness, given.

The conversion to RØD properties is:

$$A = \pi(d - t)t$$

$$J = \frac{1}{4}A ((d - t)^2 + t^2)$$

$$C = \frac{d}{2}$$

3. Coordinate system data

Given  $N_a, X_a, Y_a, Z_a, N_b, X_b, Y_b$  and  $Z_b$  and the CSTM (Coordinate System Transformation Matrices) data block, the 3 by 3 global-to-basic coordinate transformation matrices  $[T_a]$  and  $[T_b]$  are calculated using the utility routine TRANSU or TRANSS.



## MODULE FUNCTIONAL DESCRIPTIONS

### 4. Material data

Given the "MAT I.D." and  $t_\mu$ , the material routine, MAT, returns the following data:

- E - Modulus of Elasticity
- G - Shear Modulus
- $\nu$  - Poisson's ratio
- $\rho$  - Density
- $\alpha$  - Thermal expansion coefficient
- $T_0$  - Reference temperature
- $q_e$  - Structural damping ratio
- $\sigma_t$  - Stress limit, tension
- $\sigma_c$  - Stress limit, compression
- $\sigma_s$  - Stress limit, shear

### 1.87.1.2 Stiffness Matrix Calculation (Subroutines KRØD and KTUBE of Module SMA1)

1. Calculate the length of member, ( $\ell$ ):

$$\ell = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2 + (z_a - z_b)^2} \quad . \quad (1)$$

2. Calculate a normalized direction vector  $\{n\}$  in basic coordinates:

$$\begin{Bmatrix} n_1 \\ n_2 \\ n_3 \end{Bmatrix} = \frac{1}{\ell} \begin{Bmatrix} x_a - x_b \\ y_a - y_b \\ z_a - z_b \end{Bmatrix} \quad . \quad (2)$$

3. Form the extensional stiffness matrix,  $[D_\ell]$ :

$$[D_\ell] = \frac{AE}{\ell} \begin{bmatrix} n_1^2 & n_1 n_2 & n_1 n_3 \\ n_1 n_2 & n_2^2 & n_2 n_3 \\ n_1 n_3 & n_2 n_3 & n_3^2 \end{bmatrix} \quad . \quad (3)$$



# STRUCTURAL ELEMENT DESCRIPTIONS

4. Form the torsional stiffness matrix  $[D_r]$ :

$$[D_r] = \frac{GJ}{\ell} \begin{bmatrix} n_1^2 & n_1 n_2 & n_1 n_3 \\ n_1 n_2 & n_2^2 & n_2 n_3 \\ n_1 n_3 & n_2 n_3 & n_3^2 \end{bmatrix} . \quad (4)$$

5.  $[T_a]$  and  $[T_b]$  are the matrices which transform displacement components in the global coordinate system to the basic system.

6. Transforming to global coordinates and combining the results give the partitions of the element stiffness matrix:

$$[k_{aa}] = \begin{bmatrix} T_a^T D_\ell T_a & 0 \\ 0 & T_a^T D_r T_a \end{bmatrix} , \quad (5)$$

$$[k_{ab}] = - \begin{bmatrix} T_a^T D_\ell T_b & 0 \\ 0 & T_a^T D_r T_b \end{bmatrix} , \quad (6)$$

$$[k_{bb}] = \begin{bmatrix} T_b^T D_\ell T_b & 0 \\ 0 & T_b^T D_r T_b \end{bmatrix} , \quad (7)$$

$$[k_{ba}] = [k_{ab}]^T . \quad (8)$$

The element damping matrices are equal to the stiffness matrices times  $g_e$ , the structural damping coefficient.

## 4.87.1.3 Lumped Mass Matrix Calculation (Subroutines MRØD and MTUBE of Module SMA2)

The total mass of the element,  $m$ , is

$$m = \rho A \ell + \mu \ell . \quad (9)$$

# MODULE FUNCTIONAL DESCRIPTIONS

The partitions of the element mass matrix are:

$$[M_{aa}] = [M_{bb}] = \frac{m}{2} \begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ - & - & - & - & - & - \\ & & & 0 & & \\ & 0 & & & 0 & \\ & & & & & 0 \end{bmatrix}, \quad (10)$$

$$[M_{ab}] = [M_{ba}] = [0]. \quad (11)$$

## 4.87.1.4 Element Load Calculations (Subroutine ECTL of Module SSG1)

The element loading calculations are calculated using the EST data, the element loading temperature,  $\bar{T}$ , and the enforced deformation  $\delta$ .

1. Calculate  $\ell$ ,  $\{n\}$ ,  $[T_a]$  and  $[T_b]$  as in section 4.87.1.2.
2. Calculate:

$$\bar{T} = (t_a + t_b)/2 - T_0, \quad (12)$$

$$\{P_a\} = \frac{EA}{\ell} \begin{bmatrix} [T_a]^T \\ - \\ 0 \end{bmatrix} \{n\} (\delta + \alpha \ell \bar{T} - \alpha \ell T_0), \quad (13)$$

$$\{P_b\} = - \frac{EA}{\ell} \begin{bmatrix} [T_b]^T \\ - \\ 0 \end{bmatrix} \{n\} (\delta + \alpha \ell \bar{T} - \alpha \ell T_0). \quad (14)$$

3.  $\{P_a\}$  and  $\{P_b\}$  are placed in the load vector in positions corresponding to points a and b.

## 4.87.1.5 Element Stress Calculations (Subroutines SRØD1 and SRØD2 of Module SDR2).

The stress functions calculated in phase 1 (Subroutine SRØD1) are:

$$[S_a^t] = \frac{E}{\ell} \{n\}^T [T_a], \quad (1 \times 3); \quad (15)$$

# STRUCTURAL ELEMENT DESCRIPTIONS

$$[S_b^t] = - \frac{E}{\ell} \{n\}^T [T_b], \quad (1 \times 3); \quad (16)$$

$$[S_a^r] = \frac{GC}{\ell} \{n\}^T [T_b], \quad (1 \times 3); \quad (17)$$

$$[S_b^r] = - \frac{GC}{\ell} \{n\}^T [T_b], \quad (1 \times 3). \quad (18)$$

$$S_T = - \alpha E, \quad (19)$$

$$S_\delta = - \frac{E}{\ell}. \quad (20)$$

The miscellaneous constants  $A$ ,  $\frac{J}{C}$ ,  $T_0$ ,  $\sigma_t$ ,  $\sigma_c$  and  $\sigma_s$  are also saved for phase 2 calculations.

Note  $J/C$  is set to zero if  $C = 0$ . The superscripts  $t$  and  $r$  denotes translational and rotational stress matrices respectively.

The stress and force values are calculated in phase 2 (Subroutine SRØD2) using the displacement vectors  $\{u_a\}$  and  $\{u_b\}$ , the loading temperature,  $\bar{T}$ , and the enforced deformation  $\delta$ . Note that  $\{u_i^t\}$  and  $\{u_i^r\}$  ( $i = a, b$ ) denote the 3 by 1 translational and rotational components of  $\{u_i\}$ .

## 1. Partition

$$\{u_a\} \Rightarrow \begin{Bmatrix} u_a^t \\ u_a^r \end{Bmatrix}, \quad (21)$$

$$\{u_b\} \Rightarrow \begin{Bmatrix} u_b^t \\ u_b^r \end{Bmatrix}. \quad (22)$$

## 2. The stresses are:

$$\sigma = [S_a^t] \{u_a^t\} + [S_b^t] \{u_b^t\} + S_\delta \delta + S_T [\bar{T} - T_0], \quad (23)$$

$$\tau = [S_a^r] \{u_a^r\} + [S_b^r] \{u_b^r\}. \quad (24)$$

The margins of safety in tension or compression,  $M.S._t$  or  $M.S._c$  respectively, are calculated as follows:

If  $\sigma \geq 0$  then:



# MODULE FUNCTIONAL DESCRIPTIONS

$$M.S. = \begin{cases} \frac{\sigma_t}{\sigma} - 1, & \sigma_t > 0 \\ \text{Integer "1", } & \sigma_t \leq 0 \text{ or } \sigma = 0 \end{cases} \quad (25)$$

If  $\sigma < 0$ , then: define  $\sigma'_c = -|\sigma_c|$ .

$$M.S. = \begin{cases} \frac{\sigma'_c}{\sigma} - 1, & \sigma'_c \neq 0 \\ \text{Integer "1", } & \sigma = 0 \text{ or } \sigma'_c = 0 \end{cases} \quad (26)$$

The margin of safety for torsion

$$M.S. = \begin{cases} \frac{\sigma_s}{|\tau|} - 1, & \sigma_s > 0 \text{ and } \tau \neq 0 \\ \text{Integer "1", } & \sigma_s \leq 0 \text{ or } \tau = 0 \end{cases} \quad (26a)$$

The forces are:

$$P = A\sigma, \quad (27)$$

$$T = \frac{J}{C} \tau. \quad (28)$$

## 4.87.1.6 Differential Stiffness Matrix Calculation (Subroutine DRØD of Module DSMG1)

The data input from the ECPT, MPT and CSTM data blocks are as listed in section 4.87.1.1. The following variables are calculated in the same manner as those for the stiffness matrix variables (section 4.87.1.2)

$\ell$  length of rod

$\left. \begin{matrix} n_1 \\ n_2 \\ n_3 \end{matrix} \right\}$  The direction cosines of the rod axis (+ from b to a) in the basic coordinate system

$[T_a], [T_b]$  The transformation matrices from global coordinates to the basic coordinate system at the grid points.

## STRUCTURAL ELEMENT DESCRIPTIONS

Only the linear translational displacements at the grid points are extracted from the displacement vector. Call these  $\{u_a^t\}$  and  $\{u_b^t\}$  (3x1 single precision vectors.)

1. Calculate the axial load in the element (+ implies tension):

$$\frac{F_x}{\ell} = \frac{AE}{\ell^2} \left\{ \{n\}^T [T_a] \{u_a^t\} - [T_b] \{u_b^t\} \right\} - \delta - \alpha \ell \bar{T} - T_0 \quad (29)$$

2. A pair of axes perpendicular to the rod axis is constructed. Select the smallest component of  $\{n\}$ . Define  $n_i$  as the component of  $\{n\}$  which is the smallest; let  $j$  and  $k$  be the other two components. Construct  $\{m\}$  such that

$$m_i = 1, \quad m_j = m_k = 0. \quad (30)$$

Let

$$\{y\} = \frac{\{m\} \times \{n\}}{|\{m\} \times \{n\}|}, \quad (31)$$

and

$$\{z\} = \frac{\{n\} \times \{y\}}{|\{n\} \times \{y\}|}, \quad (32)$$

where  $\times$  denotes the cross product.

The actual partitions of the differential stiffness matrix relating to displacements in global coordinates are:

$$[K_{aa}^d] = \frac{F_x}{\ell} [T_a]^T [\{y\}\{y\}^T + \{z\}\{z\}^T] [T_a] \quad (33)$$

$$[K_{ab}^d] = \frac{-F_x}{\ell} [T_a]^T [\{y\}\{y\}^T + \{z\}\{z\}^T] [T_b] \quad (34)$$

$$[K_{bb}^d] = \frac{F_x}{\ell} [T_b]^T [\{y\}\{y\}^T + \{z\}\{z\}^T] [T_b] \quad (35)$$

$$[K_{ba}^d] = [K_{ab}^d]^T \quad (36)$$

# MODULE FUNCTIONAL DESCRIPTIONS

The actual 6x6 partitions are formed by expanding:

$$[k_{aa}^d] \Rightarrow \left[ \begin{array}{c|c} k_{aa}^d & 0 \\ \hline 0 & 0 \end{array} \right], \text{ etc.} \quad (37)$$

## 4.87.1.7 Piecewise Linear Analysis Calculations (Subroutine PSRØD of Module PLA3 and Subroutine PKRØD of Module PLA4)

The additional ECPTNL and ESTNL entries are:

- $\epsilon_0^*$  - The previously computed strain value once removed.
- $\epsilon^*$  - The previously computed strain value.
- $E^*$  - The previously computed modulus of elasticity.
- $T^*$  - The previously computed torsional moment (present in the ESTNL entry only).

All of the above values are initially zero with the exception of  $E^*$ , which is initially the original modulus of elasticity present on a MAT1 card.

For both stress calculation and stiffness matrix generation, the quantities  $\ell$  and  $\{n\}$  are generated as in section 4.87.1.2.

Using  $\{\Delta u_a^t\}$  and  $\{\Delta u_b^t\}$ , the 3x1 translational displacement vectors, calculate the increment of strain:

$$\Delta \epsilon = \frac{1}{\ell} \{n\}^T \left[ [T_a] \{\Delta u_a^t\} - [T_b] \{\Delta u_b^t\} \right], \quad (38)$$

$$\Delta \epsilon^* = \epsilon^* - \epsilon_0^*. \quad (39)$$

Define the following terms:

$$\epsilon_1 = \epsilon^* + \Delta \epsilon, \quad (\text{current strain}); \quad (40)$$

$$\epsilon_2 = \epsilon_1 + \gamma(\Delta \epsilon), \quad (41)$$

(estimated next strain);



# STRUCTURAL ELEMENT DESCRIPTIONS

where  $\gamma$  is the ratio of the next load increment to the present load increment.

We calculate:

$$\sigma_1 = f(\epsilon_1), \quad (42)$$

$$\sigma_2 = f(\epsilon_2), \quad (43)$$

where  $f$  is the tabular stress-strain function. (When  $\epsilon^* = 0$ , define  $\sigma_1 = E_0 \epsilon_1$ )

For stiffness matrix generation, the new material properties are:

$$E = \begin{cases} \frac{\sigma_2 - \sigma_1}{\epsilon_2 - \epsilon_1}, & \text{if } \epsilon_2 \neq \epsilon_1 \\ E^*, & \text{if } \epsilon_2 = \epsilon_1 \end{cases}; \quad (44)$$

and

$$G = \frac{E}{E_0} G_0, \quad (45)$$

where  $E_0$  and  $G_0$  are elastic moduli obtained from the MAT1 bulk data card via subroutine MAT.

For plastic element stresses and forces the values are:

$$\sigma = \sigma_1, \quad (46)$$

$$P = A\alpha_1, \quad (47)$$

$$T = \frac{JG^*}{\ell} \{n\}^T ([T_a] \{\Delta u_a^r\} - [T_b] \{\Delta u_b^r\}) + T^*, \quad (48)$$

$$\tau = \frac{CT}{J} \quad (\tau = 0 \text{ if } C = 0 \text{ or } J = 0), \quad (49)$$

where

$$G^* = \frac{E^*}{E_0} G_0. \quad (50)$$

# MODULE FUNCTIONAL DESCRIPTIONS

The new ESTNL and ECPTNL entries are:

$$\epsilon_{on}^* = \epsilon^* , \quad (51)$$

$$\epsilon_n^* = \epsilon_1 , \quad (52)$$

$$E_n^* = \frac{\sigma_2 - \sigma_1}{\epsilon_2 - \epsilon_1} , \quad (53)$$

$$T_n^* = T . \quad (54)$$

## STRUCTURAL ELEMENT DESCRIPTIONS

### 4.87.1.8 Coupled Mass Matrix Calculation (Subroutine MCRØD of Module SMA2)

1. The length of the element,  $\ell$ , the normalized direction vector,  $\{n\}$ , and the mass of the element,  $m$ , are calculated as in Equations 1 and 2 in Section 4.87.1.2 and Equation 9 in Section 4.87.1.3, respectively.

2. The 3 by 3 matrix

$$[\Delta M] = \frac{m}{12} \begin{bmatrix} n_1^2 & n_1 n_2 & n_1 n_3 \\ n_1 n_2 & n_2^2 & n_2 n_3 \\ n_1 n_3 & n_2 n_3 & n_3^2 \end{bmatrix} \quad (55)$$

is calculated.

3. The 3 by 3 element mass matrices in basic coordinates are

$$[m_{aa}] = [m_{bb}] = \begin{bmatrix} m/2 & 0 & 0 \\ 0 & m/2 & 0 \\ 0 & 0 & m/2 \end{bmatrix} - [\Delta M], \quad (56)$$

and

$$[m_{ab}] = [m_{ba}] = [\Delta M]. \quad (57)$$

4. In global coordinates, the 6 by 6 mass matrix partitions are:

$$[M_{ij}] = \begin{bmatrix} T_i^T m_{ij} T_j & 0 \\ 0 & 0 \end{bmatrix}, \quad (58)$$

for  $i = a$  or  $b$ ,  $j = a$  or  $b$ , and where  $[T_i]$  is the global-to-basic coordinate transformation matrix for point  $i$ .

### 4.87.1.9 Thermal Analysis Calculations for the RØD Elements

If a "stiffness" matrix for thermal analysis is to be generated in subroutine KRØD, word 56 in CØMMØN data block SYSTEM is +1. The length,  $\ell$ , of the element is calculated as in Section 4.87.1.2. The thermal material coefficient,  $k$ , is obtained by calling subroutine HMAT, rather than MAT. The matrix terms are:



# MODULE FUNCTIONAL DESCRIPTIONS

For the pivot point,  $i = 1$  or  $2$ :

$$K_{ii} = \frac{k A}{\ell}.$$

For  $j \neq i$ ,

$$K_{ij} = - \frac{k A}{\ell}.$$

The "mass" matrix for thermal analysis is calculated in subroutine MRØD. The thermal capacity,  $C_p$ , is given by subroutine HMAT. The matrix terms, to be placed in matrix BGG, are:

$$B_{ii} = \frac{C_p A \ell}{2} \quad i = 1, 2.$$

The heat flux and gradient "stress" recovery is performed by subroutines SDHTF1, SDHTFF, and SDHTF2 in module SDR2. In Phase 1, the value  $H$  is extracted with subroutine HMAT and the output is:

$$K_1 = H$$

$$[C] = \frac{1}{\ell} [-1 \ 1].$$

In Phase 2, the gradient and flux are:

$$\Delta T = [C] \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix}$$

$$Q = - K_1 \Delta T$$

## STRUCTURAL ELEMENT DESCRIPTIONS

### 4.87.2 The BAR Element

#### 4.87.2.1 Input Data for the BAR Element

1. The ECPT/EST entries for the BAR are:

| <u>Symbol</u>                                      | <u>Descriptions</u>   |
|--|---|
| $SIL_a, SIL_b$                                     | Scalar indices of grid points a and b   |
| $N_a, X_a, Y_a, Z_a \}$<br>$N_b, X_b, Y_b, Z_b \}$ | Local coordinate system number and location in basic coordinates of the grid points                                   |
| $x_1, x_2, x_3$                                    | Orientation vector (see Figure 1 in section 1.3 of the User's Manual)   |
| F  | Flag for orientation vector definition  |
| $P_a, P_b$   | Pin flags for either end  |
| Mat I. D.  | Material property identification number   |
| A  | Cross-sectional area  |
| $I_1, I_2$   | Bending inertials in element coordinates about axes normal to reference planes 1 and 2 respectively                   |
| $I_{12}$   | Cross-product bending inertia   |
| J  | Torsional Inertia   |
| $\mu$  | Nonstructural mass per unit length  |
| $a_x, a_y, a_z \}$<br>$b_x, b_y, b_z \}$           | Vectors defining offset distances between BAR ends and grid points (see Figure 1 in section 1.3 of the User's Manual) |
| $K_1, K_2$   | Shear factors   |
| $c_1, c_2, d_1, d_2 \}$<br>$f_1, f_2, g_1, g_2 \}$ | Positions on cross section of four points for stress calculations (see section 1.3.2 of the User's Manual)            |
| $t_\mu$  | Temperature for material properties   |

2. Coordinate system data

The location ( $X_i, Y_i, Z_i$ ) and local coordinate system number ( $N_i$ ) of each grid point ( $i = a$  or  $b$ ) are used to calculate the 3 by 3 global-to-basic coordinate transformation matrices,  $[T_a]$  and  $[T_b]$ .

## MODULE FUNCTIONAL DESCRIPTIONS

### 3. Material data

The material identification number "Mat I.D." and  $t_\mu$  are used to select the following:

|            |   |                               |
|------------|---|-------------------------------|
| $E$        | - | Modulus of elasticity         |
| $G$        | - | Shear modulus                 |
| $\nu$      | - | Poisson's ratio               |
| $\rho$     | - | Density                       |
| $\alpha$   | - | Thermal expansion coefficient |
| $T_0$      | - | Reference temperature         |
| $g_e$      | - | Structural damping ratio      |
| $\sigma_t$ | - | Stress limit, tension         |
| $\sigma_c$ | - | Stress limit, compression     |
| $\sigma_s$ | - | Stress limit, shear           |

#### 4.87.2.2 Stiffness Matrix Calculation (Subroutine KBAR of Module SMA1)

1. If the orientation flag  $F$  is nonzero, transform the given vector to basic coordinates:

$$\{v_0\} = [T_a] \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix}. \quad (1)$$

Otherwise,

$$\{v_0\} = \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix}. \quad (2)$$

2. Transfer the relative beam end locations to basic coordinates:

$$\begin{Bmatrix} \delta_{a1} \\ \delta_{a2} \\ \delta_{a3} \end{Bmatrix} = [T_a] \begin{Bmatrix} a_x \\ a_y \\ a_z \end{Bmatrix}, \quad (3)$$



# STRUCTURAL ELEMENT DESCRIPTIONS

$$\begin{Bmatrix} \delta_{b1} \\ \delta_{b2} \\ \delta_{b3} \end{Bmatrix} = [T_b] \begin{Bmatrix} b_x \\ b_y \\ b_z \end{Bmatrix} . \quad (4)$$

3. The center axis of the beam, defined as {i} is calculated as:

$$\{V_i\} = - \begin{Bmatrix} X_a - X_b + \delta_{a1} - \delta_{b1} \\ Y_a - Y_b + \delta_{a2} - \delta_{b2} \\ Z_a - Z_b + \delta_{a3} - \delta_{b3} \end{Bmatrix} , \quad (5)$$

$$\rho = (V_{i1}^2 + V_{i2}^2 + V_{i3}^2)^{1/2} , \quad (6)$$

$$\{i\} = \frac{1}{\rho} \{V_i\} . \quad (7)$$

4. The bending axis of the beam in plane 2 is:

$$\{k\} = \frac{\{i\} \times \{v_o\}}{|\{i\} \times \{v_o\}|} . \quad (8)$$

5. The bending axis of the beam in plane 1 is:

$$\{j\} = \frac{\{k\} \times \{i\}}{|\{k\} \times \{i\}|} . \quad (9)$$

6. The 6x6 matrix for transforming element displacements in the element coordinates to basic coordinate displacements is:

$$[T_{eb}] = \begin{bmatrix} \{i\} & \{j\} & \{k\} & 0 & 0 & 0 \\ 0 & 0 & 0 & \{i\} & \{j\} & \{k\} \end{bmatrix} . \quad (10)$$

7. The 6 by 6 matrices for transforming global coordinate displacements to basic coordinate displacements are:

$$[C_a] = \begin{bmatrix} T_a & 0 \\ 0 & T_a \end{bmatrix} , \quad (11)$$

# MODULE FUNCTIONAL DESCRIPTIONS

$$[C_b] = \begin{bmatrix} T_b & 0 \\ -T_b & 0 \\ 0 & T_b \end{bmatrix} \quad (12)$$

8. The 6 by 6 matrices for transforming displacements of the grid points to displacements of the element ends are:

$$[E_a] = \begin{bmatrix} 1 & 0 & 0 & 0 & a_z & -a_y \\ 0 & 1 & 0 & -a_z & 0 & a_x \\ 0 & 0 & 1 & a_y & -a_x & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (13)$$

$$[E_b] = \begin{bmatrix} 1 & 0 & 0 & 0 & b_z & -b_y \\ 0 & 1 & 0 & -b_z & 0 & b_x \\ 0 & 0 & 1 & b_y & -b_x & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (14)$$

# STRUCTURAL ELEMENT DESCRIPTIONS

9. The 6 by 6 partitions of the element stiffness matrix in element coordinates are:

$$[K_{aa}^e] = \begin{bmatrix} \frac{AE}{l} & 0 & 0 & 0 & 0 & 0 \\ 0 & R_1 & \beta & 0 & -\frac{l}{2}\beta & \frac{l}{2}R_1 \\ 0 & \beta & R_2 & 0 & -\frac{l}{2}R_2 & \frac{l}{2}\beta \\ 0 & 0 & 0 & \frac{GJ}{l} & 0 & 0 \\ 0 & -\frac{l}{2}\beta & -\frac{l}{2}R_2 & 0 & k_2 & -\frac{l^2}{3}\beta \\ 0 & \frac{l}{2}R_1 & \frac{l}{2}\beta & 0 & -\frac{l^2}{3}\beta & k_1 \end{bmatrix}, \quad (15)$$

$$[K_{ab}^e] = \begin{bmatrix} -\frac{AE}{l} & 0 & 0 & 0 & 0 & 0 \\ 0 & -R_1 & -\beta & 0 & -\frac{l}{2}\beta & \frac{l}{2}R_1 \\ 0 & -\beta & -R_2 & 0 & -\frac{l}{2}R_2 & \frac{l}{2}\beta \\ 0 & 0 & 0 & -\frac{GJ}{l} & 0 & 0 \\ 0 & \frac{l}{2}\beta & \frac{l}{2}R_2 & 0 & k_4 & -\frac{l^2}{6}\beta \\ 0 & -\frac{l}{2}R_1 & -\frac{l}{2}\beta & 0 & -\frac{l^2}{6}\beta & k_3 \end{bmatrix}, \quad (16)$$

$$[K_{ba}^e] = [K_{ab}^e]^T, \quad (17)$$



# MODULE FUNCTIONAL DESCRIPTIONS

$$[K_{bb}^e] = \begin{bmatrix} \frac{AE}{\ell} & 0 & 0 & 0 & 0 & 0 \\ 0 & R_1 & \beta & 0 & \frac{\ell}{2}\beta & -\frac{\ell}{2}R_1 \\ 0 & \beta & R_2 & 0 & \frac{\ell}{2}R_2 & -\frac{\ell}{2}\beta \\ 0 & 0 & 0 & \frac{GJ}{\ell} & 0 & 0 \\ 0 & \frac{\ell}{2}\beta & \frac{\ell}{2}R_2 & 0 & k_2 & -\frac{\ell^2}{3}\beta \\ 0 & -\frac{\ell}{2}R_1 & -\frac{\ell}{2}\beta & 0 & -\frac{\ell^2}{3}\beta & k_1 \end{bmatrix} \quad (18)$$

The terms are defined as:

If  $I_{12} = 0$ :

$$\beta = 0, \quad (19)$$

$$R_1 = \frac{12EI_1}{\ell^3} \left[ 1 + \frac{12EI_1}{K_1AG\ell^2} \right]^{-1}, \quad (20)$$

$$R_2 = \frac{12EI_2}{\ell^3} \left[ 1 + \frac{12EI_2}{K_2AG\ell^2} \right]^{-1}. \quad (21)$$

Note: If  $K_i A G = 0$ , set  $\frac{1}{K_i A G} = 0$ ,  $i = 1$  or  $2$

If  $I_{12} \neq 0$

$$R_1 = \frac{12EI_1}{\ell^3}, \quad (22)$$

$$R_2 = \frac{12EI_2}{\ell^3}, \quad (23)$$

$$\beta = \frac{12EI_{12}}{\ell^3}. \quad (24)$$

# STRUCTURAL ELEMENT DESCRIPTIONS

Note: In this case no shearing deformations are calculated.

For both cases

$$k_1 = \frac{\ell^2}{4} R_1 + \frac{EI_1}{\ell}, \quad (25)$$

$$k_2 = \frac{\ell^2}{4} R_2 + \frac{EI_2}{\ell}, \quad (26)$$

$$k_3 = \frac{\ell^2}{4} R_1 - \frac{EI_1}{\ell}, \quad (27)$$

$$k_4 = \frac{\ell^2}{4} R_2 - \frac{EI_2}{\ell}. \quad (28)$$

10. Process the end condition ("pin") data. The nonzero digits of the "pin flag" integers  $P_a$  and  $P_b$  specify the following:

$$P = \left\{ \begin{array}{l} 1 \text{ implies no forces are transmitted to the element in the x-direction at the pinned end} \\ 2 \text{ implies no forces are transmitted to the element in the y-direction at the pinned end} \\ 3 \text{ implies no forces are transmitted to the element in the z-direction at the pinned end} \\ 4 \text{ implies no forces are transmitted to the element in the } \theta_x \text{-direction at the pinned end} \\ 5 \text{ implies no forces are transmitted to the element in the } \theta_y \text{-direction at the pinned end} \\ 6 \text{ implies no forces are transmitted to the element in the } \theta_z \text{-direction at the pinned end} \end{array} \right.$$

1) Nonzero digits of the number  $P$  specify the unconnected degrees of freedom on the end of the BAR.

2) Construct the overall element matrix and perform the following operations:

a)

$$\left[ \begin{array}{c|c} k_{aa}^e & k_{ab}^e \\ \hline k_{ba}^e & k_{bb}^e \end{array} \right] = \left[ \begin{array}{ccc} k_{11} & k_{12} & \dots \\ k_{21} & & \\ \vdots & & k_{12,12} \end{array} \right]. \quad (29)$$

# MODULE FUNCTIONAL DESCRIPTIONS

b) Convert the pin numbers to row numbers in the  $[k]$  matrix. If a pin number refers to end "a", it corresponds to the row number. If it refers to end "b", the row number is obtained by adding six to the pin number.

c) For each row of the  $[k]$  matrix perform the following operation to obtain the new stiffness matrix  $[k']$

$$k'_{j\ell} = k_{j\ell} - \frac{k_{i\ell}k_{ji}}{k_{ii}} \quad \begin{cases} j = 1, \dots, 12, j \neq i \\ \ell = 1, \dots, 12, \ell \neq i \end{cases}, \quad (30)$$

and  $k'_{j\ell} = 0$  for  $j = i$  or  $\ell = i$ , where  $i$  is the row number obtained from the pin number as in b).

This operation causes the  $i^{\text{th}}$  row and column to be zero, and disconnects that degree of freedom from the matrix. Repeat for each pin index.

d) Repartition the matrix into the four original sections, carrying the zero rows and columns along.

11. The equations to convert the partitions to global coordinates are:

$$[k_{aa}] = \{T_{eb}^T C_a E_a\}^T [k_{aa}^e] \{T_{eb}^T C_a E_a\}, \quad (31)$$

$$[k_{ab}] = \{T_{eb}^T C_a E_a\}^T [k_{ab}^e] \{T_{eb}^T C_b E_b\}, \quad (32)$$

$$[k_{bb}] = \{T_{eb}^T C_b E_b\}^T [k_{bb}^e] \{T_{eb}^T C_b E_b\}, \quad (33)$$

$$[k_{ba}] = [k_{ab}]^T. \quad (34)$$



# STRUCTURAL ELEMENT DESCRIPTIONS

## 4.87.2.3 Lumped Mass Matrix Calculation (Subroutine MBAR of Module SMA2)

$$[M_a] = [M_b] = \begin{bmatrix} m/2 & & \\ & m/2 & 0 \\ & & m/2 & \\ - & - & - & - \\ & 0 & & 0 \end{bmatrix}, \quad (35)$$

where:

$$m = \ell (\rho A + \mu), \quad (36)$$

and:

$$[M_{aa}] = \{T_{eb}^T C_a E_a\}^T [M_a] \{T_{eb}^T C_a E_a\}, \quad (37)$$

$$[M_{bb}] = \{T_{eb}^T C_b E_b\}^T [M_b] \{T_{eb}^T C_b E_b\}, \quad (38)$$

$$[M_{ab}] = [M_{ba}] = [0]. \quad (39)$$

The equations for the generation of the "consistent" or coupled mass matrix for the BAR are given in section 4.87.2.8.

## 4.87.2.4 Element Load Calculation (Subroutine BAR of Module SSG1)

a) Form  $\ell$ ,  $\{i\}$ ,  $[C_a]$ ,  $[C_n]$ ,  $[E_a]$ ,  $[E_b]$ , and  $[k']$  as in Equation 30.

b) Partition the  $12 \times 12$  matrix into four  $6 \times 6$  matrices

$$[k'] \Rightarrow \begin{bmatrix} K_{aa}^e & K_{ab}^e \\ K_{ba}^e & K_{bb}^e \end{bmatrix}. \quad (40)$$

c) Form the vector

$$\{u_a^t\} = \begin{Bmatrix} -\alpha\ell(\bar{T} - T_0) - \delta \\ -\frac{\alpha\ell^2}{6} [T'_{1a} + 2T'_{1b}] \\ -\frac{\alpha\ell^2}{6} [T'_{2a} + 2T'_{2b}] \\ 0 \\ -\frac{\alpha\ell}{2} [T'_{2a} + T'_{2b}] \\ \frac{\alpha\ell}{2} [T'_{1a} + T'_{1b}] \end{Bmatrix}, \quad (41)$$

where  $\bar{T}$  is the average of  $\bar{T}_a$  and  $\bar{T}_b$ ,  $\delta$  is the enforced deformation, and  $T'_i$  are the gradients.

d) The load vectors in global coordinates are:

$$\begin{aligned} \{P_a\} &= [E_a^T][C_a^T][T_{eb}][K_{aa}^P]\{u_a^t\}, \\ \{P_b\} &= [E_b^T][C_b^T][T_{eb}][K_{ba}^P]\{u_a^t\}. \end{aligned} \quad (42)$$

## MODULE FUNCTIONAL DESCRIPTIONS

### 4.87.2.5 Element Stress Calculations (Subroutines SBAR1 and SBAR2 of Module SDR2)

The stress and force data are calculated in two phases. The first phase (subroutine SBAR1) calculates unique stress versus displacement, temperature and enforced deformation functions for each element. The second phase (subroutine SBAR2) applies the various subcase displacement vectors to product the element forces and stresses.

Phase 1 calculations are as follows:

1. Using the algorithms given in the description of the stiffness matrix calculations for the element (Section 4.87.2.2), calculate the following data:

$[T_{eb}]$  = 6x6 element coordinate transformation

$[E_a], [E_b]$  - Offset transformation matrices (6x6)

$[C_a], [C_b]$  - 6x6 global to basic coordinate transformations

$[K_e']$  - 12x12 stiffness matrix in element coordinates with pin joint effects

$l$  - Length of BAR

2. Partition the stiffness matrix  $[K_e']$  saving only the upper 6x6 matrices,  $[k_{aa}]$  and  $[k_{ab}]$ .

$$[K_e'] \Rightarrow \begin{bmatrix} k_{aa} & | & k_{ab} \\ \hline & & \\ k_{ba} & | & k_{bb} \\ \hline & & \end{bmatrix} \quad (43)$$

3. The stress matrices are:

$$[S_a] = [k_{aa}][T_{eb}]^T[C_a][E_a] , \quad (44)$$

$$[S_b] = [k_{ab}][T_{eb}]^T[C_b][E_b] . \quad (45)$$



# STRUCTURAL ELEMENT DESCRIPTIONS

4. The temperature and enforced deformation matrix is:

$$[S_t] = [K_{aa}] \begin{bmatrix} \alpha l & 0 & 0 & 0 & 0 \\ 0 & \frac{\alpha l^2}{6} & 2\frac{\alpha l^2}{6} & 0 & 0 \\ 0 & 0 & 0 & \frac{\alpha l^2}{6} & 2\frac{\alpha l^2}{6} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\alpha l}{2} & \frac{\alpha l}{2} \\ 0 & -\frac{\alpha l}{2} & -\frac{\alpha l}{2} & 0 & 0 \end{bmatrix} \quad (46)$$

Phase 2 element force calculations are as follows:

1. The static element forces are calculated by the equation:

$$\{P\} = [S_a]\{u_a\} + [S_b]\{u_b\} + [S_t] \begin{Bmatrix} \bar{T}_z \\ T'_{1a} \\ T'_{1b} \\ T'_{2a} \\ T'_{2b} \end{Bmatrix}, \quad (47)$$

where  $[S_a]$  and  $[S_b]$  are the displacement-stress matrices,  $\{u_a\}$  and  $\{u_b\}$  are the displacement vectors, and  $\bar{T}_z$ ,  $T'_{1a}$ , etc. are the element thermal resultants. In terms of the given temperatures at the ends,  $\bar{T}_a$  and  $\bar{T}_b$ , the equation for  $\bar{T}_z$  is:

$$\bar{T}_z = \frac{\bar{T}_a + \bar{T}_b}{2} - T_0 \quad (48)$$

2. The element axial force is:

$$F_x = -[P_1 + S_\delta \delta] \quad (49)$$

# MODULE FUNCTIONAL DESCRIPTIONS

3. The element shear loads are:

$$V_1 = -P_2 , \quad (51)$$

$$V_2 = -P_3 . \quad (52)$$

4. The torque and moments are:

$$T = -P_4 , \quad (53)$$

$$M_{1a} = -P_6 , \quad (54)$$

$$M_{2a} = P_5 , \quad (55)$$

$$M_{1b} = M_{1a} - V_1 \ell , \quad (56)$$

$$M_{2b} = M_{2a} - V_2 \ell . \quad (57)$$

Phase 2 element stress calculations are as follows:

1. The stresses due to bending are:

$$k_{1a} = \frac{M_{2a} I_{12} - M_{1a} I_2}{I_1 I_2 - I_{12}^2} , \quad (58)$$

$$k_{2a} = \frac{M_{1a} I_{12} - M_{2a} I_1}{I_1 I_2 - I_{12}^2} , \quad (59)$$

$$\sigma_{ca} = k_{1a} c_1 + k_{2a} c_2 , \quad (60)$$

$$\sigma_{da} = k_{1a} d_1 + k_{2a} d_2 , \quad (61)$$

$$\sigma_{fa} = k_{1a} f_1 + k_{2a} f_2 , \quad (62)$$

$$\sigma_{ga} = k_{1a} g_1 + k_{2a} g_2 . \quad (63)$$

For  $\sigma_{cb}$ ,  $\sigma_{db}$ ,  $\sigma_{fb}$ ,  $\sigma_{gb}$  use the above equations interchanging the subscripts for b and a.

---

Equation 50 is intentionally missing.

# MODULE FUNCTIONAL DESCRIPTIONS

The stresses calculated at points c, d, e, and f on the cross section will be modified by the element temperatures  $T_{ac}$ ,  $T_{ad}$ , ...  $T_{bc}$ , ... if at least one of the  $T$  values is nonzero. At end a:

$$\Delta\sigma_c = -E\alpha(T_{ac} - T'_{1a} c_1 - T'_{2a} c_2 - \bar{T}_a)$$

$$\Delta\sigma_d = -E\alpha(T_{ad} - T'_{1a} d_1 - T'_{2a} d_2 - \bar{T}_a)$$

etc.

.

.

.

At end b:

$$\Delta\sigma_c = -E\alpha(T_{bc} - T'_{1b} c_1 - T'_{2b} c_2 - \bar{T}_b)$$

$$\Delta\sigma_d = -E\alpha(T_{bd} - T'_{1b} d_1 - T'_{2b} d_2 - \bar{T}_b)$$

etc.

.

.

.

(63a)

( $T_a$  and  $T_b$  are the given average temperatures at the ends.) The above stresses are added to the stresses calculated in Equations 60 - 63.

2. The axial stress is:

$$\sigma_{ax} = \frac{F_x}{A} \quad (64)$$

3. The maxima and minima are:

$$\sigma_{a \max} = \sigma_{ax} + \max(\sigma_{ca}, \sigma_{da}, \sigma_{fa}, \sigma_{ga}) \quad (65)$$

$$\sigma_{b \max} = \sigma_{ax} + \max(\sigma_{cb}, \sigma_{db}, \sigma_{fb}, \sigma_{gb}) \quad (66)$$

$$\sigma_{a \min} = \sigma_{ax} + \min(\sigma_{ca}, \sigma_{da}, \sigma_{fa}, \sigma_{ga}) \quad (67)$$

$$\sigma_{b \min} = \sigma_{ax} + \min(\sigma_{cb}, \sigma_{db}, \sigma_{fb}, \sigma_{gb}) \quad (68)$$



## STRUCTURAL ELEMENT DESCRIPTIONS

4. The margins of safety in tension,  $M.S._t$ , and compression,  $M.S._c$ , are as follows:

$$M.S._t = \begin{cases} \min \left( \frac{\sigma_t}{\sigma_{a \max}}, \frac{\sigma_t}{\sigma_{b \max}} \right) - 1, & \sigma_t > 0 \\ \text{Integer "1"} & , \sigma_t \leq 0 \text{ or } \max(\sigma_{a \max}, \sigma_{b \max}) \leq 0 \end{cases} \quad (69)$$

Define  $\sigma'_c = -|\sigma_c|$ . Then:

$$M.S._c = \begin{cases} \min \left( \frac{\sigma'_c}{\sigma_{a \min}}, \frac{\sigma'_c}{\sigma_{b \min}} \right) - 1, & \sigma'_c \neq 0 \\ \text{Integer "1"} & , \sigma'_c = 0 \text{ or } \min(\sigma_{a \min}, \sigma_{b \min}) \geq 0 \end{cases} \quad (70)$$

### 4.87.2.6 Differential Stiffness Matrix Calculation (Subroutine DBEAM of Module DSMG1)

Many of the equations used in this calculation routine are identical to the stiffness matrix and element force calculations. Refer to sections 4.87.2.2 and 4.87.2.5 for details.

1. Calculate  $[T_{eb}]$ ,  $[C_a]$ ,  $[C_b]$ ,  $[E_a]$ ,  $[E_b]$  and  $[K]$ , the matrices used in the BAR stiffness matrix generation, section 4.87.2.2.
2. Calculate the forces in the element using the equations in section 4.87.2.5.
3. The number 2, 3, 4, 5, 6, 8, 9, 10, 11 and 12<sup>th</sup> rows and columns of the 12 by 12 differential stiffness matrix are given in Figure 1. The first and seventh rows and columns are zero. The terms are identical to the element forces calculated for output (Equations 51, 52 and 54 through 57 of section 4.87.2.5) with the following notational changes:

$$\begin{array}{ll} M_{ay} = M_{2a} & M_{by} = M_{2b} \\ M_{az} = M_{1a} & M_{bz} = M_{1b} \\ V_y = V_1 & V_z = V_2 \end{array}$$

$[K_e^d] =$

|                    |                    |                       |                    |                      |                     |                     |                       |                    |                    |
|--------------------|--------------------|-----------------------|--------------------|----------------------|---------------------|---------------------|-----------------------|--------------------|--------------------|
| $\frac{6F_x}{5l}$  | 0                  | $\frac{M_{by}}{l}$    | 0                  | $-\frac{F_x}{10}$    | $-\frac{6F_x}{5l}$  | 0                   | $\frac{M_{ay}}{l}$    | 0                  | $-\frac{F_x}{10}$  |
| 0                  | $\frac{6F_x}{5l}$  | $\frac{M_{bz}}{l}$    | $\frac{F_x}{10}$   | 0                    | 0                   | $-\frac{6F_x}{5l}$  | $\frac{M_{az}}{l}$    | $\frac{F_x}{10}$   | 0                  |
| $\frac{M_{by}}{l}$ | $\frac{M_{bz}}{l}$ | $\frac{JF_x}{l^2 A}$  | $-\frac{lV_y}{6}$  | $-\frac{lV_z}{6}$    | $-\frac{M_{by}}{l}$ | $-\frac{M_{bz}}{l}$ | $-\frac{JF_x}{l^2 A}$ | $\frac{lV_y}{6}$   | $\frac{lV_z}{6}$   |
| 0                  | $\frac{F_x}{10}$   | $-\frac{lV_y}{6}$     | $\frac{2lF_x}{15}$ | 0                    | 0                   | $-\frac{F_x}{10}$   | $\frac{lV_y}{6}$      | $-\frac{lF_x}{30}$ | 0                  |
| $-\frac{F_x}{10}$  | 0                  | $-\frac{lV_z}{6}$     | 0                  | $-\frac{2l}{15} F_x$ | $\frac{F_x}{10}$    | 0                   | $\frac{lV_z}{6}$      | 0                  | $-\frac{lF_x}{30}$ |
| $-\frac{6F_x}{5}$  | 0                  | $-\frac{M_{by}}{l}$   | 0                  | $\frac{F_x}{10}$     | $\frac{6F_x}{5l}$   | 0                   | $-\frac{M_{ay}}{l}$   | 0                  | $\frac{F_x}{10}$   |
| 0                  | $-\frac{6F_x}{l}$  | $-\frac{M_{bz}}{l}$   | $-\frac{F_x}{10}$  | 0                    | 0                   | $\frac{6F_x}{5l}$   | $-\frac{M_{az}}{l}$   | $-\frac{F_x}{10}$  | 0                  |
| $\frac{M_{ay}}{l}$ | $\frac{M_{az}}{l}$ | $-\frac{JF_x}{l^2 A}$ | $\frac{lV_y}{6}$   | $\frac{lV_z}{6}$     | $-\frac{M_{ay}}{l}$ | $-\frac{M_{az}}{l}$ | $\frac{JF_x}{l^2 A}$  | $-\frac{lV_y}{6}$  | $-\frac{lV_z}{6}$  |
| 0                  | $\frac{F_x}{10}$   | $\frac{lV_y}{6}$      | $-\frac{lF_x}{30}$ | 0                    | 0                   | $-\frac{F_x}{10}$   | $-\frac{lV_y}{6}$     | $\frac{2lF_x}{15}$ | 0                  |
| $-\frac{F_x}{10}$  | 0                  | $\frac{lV_z}{6}$      | 0                  | $-\frac{lF_x}{30}$   | $\frac{F_x}{10}$    | 0                   | $-\frac{lV_z}{6}$     | 0                  | $\frac{2lF_x}{15}$ |

Figure 1. - Differential stiffness matrix for a BAR element, rows and columns 1 and 7 deleted.

# STRUCTURAL ELEMENT DESCRIPTIONS

4. The effects of "pin joints" are added by applying the elastic stiffness constraints. The elastic stiffness matrix,  $[K^e]$ , with no pin joints is equivalent to the matrix  $[k]$  in Equation 29. If coordinate number  $j$  is released by a pin flag the differential stiffness matrix must be modified as follows:

a. If  $i \neq j$  and  $\ell \neq j$ ,  $i = 1, 2, \dots, 12$  and  $\ell = 1, 2, \dots, 12$ :

$$(K_{i\ell}^{d*})_m = (K_{i\ell}^{d*})_{m-1} - \frac{K_{\ell j}^e (K_{ji}^{d*})_{m-1}}{K_{jj}^e} - \frac{K_{ji}^e (K_{\ell j}^{d*})_{m-1}}{K_{jj}^e} + \frac{K_{\ell j}^e K_{ji}^e + (K_{jj}^{d*})_{m-1}}{(K_{jj}^e)^2}, \quad (71)$$

where  $m$  is the (row) index of the pin joint number,  $1 \leq m \leq 12$ . For  $m = 0$ , define

$$(K_{i\ell}^{d*})_0 = (K_{i\ell}^d). \quad (72)$$

b. If  $i$  or  $j = \ell$

$$K_{ij}^{d*} = 0 \quad i = 1, \dots, 12, \quad (73)$$

$$K_{j\ell}^{d*} = 0 \quad j = 1, \dots, 12. \quad (74)$$

5. The 12 by 12 matrix  $[K^{d*}]$  is now partitioned into 6 by 6 matrices related to each grid point

$$[K^{d*}] = \begin{bmatrix} K_{aa}^{d*} & \vdots & K_{ab}^{d*} \\ \vdots & \ddots & \vdots \\ K_{ba}^{d*} & \vdots & K_{bb}^{d*} \end{bmatrix}. \quad (75)$$

6. If point "p" is the pivot point ( $p = a$  or  $b$ ), the matrices generated in global coordinates are:

$$[K_{pa}^d] = ([T_{eb}]^T [C_p] [E_p])^T [K_{pa}^{d*}] ([T_{eb}] [C_a] [E_a]), \quad (76)$$

$$[K_{pb}^d] = ([T_{eb}]^T [C_p] [E_p])^T [K_{pb}^{d*}] ([T_{eb}]^T [C_b] [E_b]). \quad (77)$$



## MODULE FUNCTIONAL DESCRIPTIONS

### 4.87.2.7 Piecewise Linear Analysis Calculations (Subroutine PSBAR of Module PLA3 and Subroutine PKBAR of Module PLA4)

The additional ECPTNL and ESTNL data block entries for a BAR element are:

|                |  |
|----------------|--|
| $\epsilon_0^*$ | - The previously computed axial strain value once removed. |
| $\epsilon^*$   | - The previously computed axial strain value.              |
| $E^*$          | - The previously computed modulus of elasticity.           |
| $V_1^*$        | - The previously computed element forces and moments.      |
| $V_2^*$        |  |
| $T^*$          |  |
| $M_{1a}^*$     |  |
| $M_{2a}^*$     |  |

All of the above values are initially zero with the exception of  $E^*$ , which is initially the original modulus of elasticity present on a MAT1 bulk data card.

For both stress (subroutine PSBAR) and stiffness matrix (subroutine PKBAR) calculations, the following data are generated:

$\lambda$ ,  $[T_{eb}]$ ,  $[C_a]$ ,  $[C_b]$ ,  $[E_a]$ ,  $[E_b]$ ,  $[k_{aa}^e]$ , and  $[k_{ab}^e]$  as in Equations 6, 10, 11, 12, 13, 14 and 30 in section 4.87.2.1. Note that: a)  $[k_{aa}^e]$  and  $[k_{ab}^e]$  are the partitions of the stiffness matrix with pin joint effects taken into account; b) for stress calculations,  $E^*$  is used to compute  $[k_{aa}^e]$  and  $[k_{ab}^e]$ ; and c) for stiffness matrix calculations  $E_1$  (see Equation 84 below) is used to compute  $[k_{aa}^e]$  and  $[k_{ab}^e]$ .

Using the incremental displacement vectors,  $\{\Delta u_a\}$  and  $\{\Delta u_b\}$ , calculate the incremental strain:

$$\Delta \epsilon = \frac{1}{\lambda} \{T_{eb1}\}^T \left[ [C_b][E_b]\{\Delta u_b\} - [C_a][E_a]\{\Delta u_a\} \right], \quad (78)$$

where  $\{T_{eb1}\}$  is the first column of  $[T_{eb}]$ . If coordinate "1" of either  $P_a$  or  $P_b$  (the pin flags) is "on", the element is treated as linear. This determination is made in the Piecewise Linear

# STRUCTURAL ELEMENT DESCRIPTIONS

Analysis pre-processor module, PLAI.

Calculate the extensional strains:

$$\Delta \epsilon^* = \epsilon^* - \epsilon_0^* , \quad (79)$$

$$\epsilon_1 = \epsilon^* + \Delta \epsilon , \quad (80)$$

$$\epsilon_2 = \epsilon_1 + \gamma(\Delta \epsilon) , \quad (81)$$

where  $\gamma$  is the ratio of the next load increment to the present load increment.

The stresses

$$\sigma_1 = f(\epsilon_1) , \quad (82)$$

$$\sigma_2 = f(\epsilon_2) , \quad (83)$$

are computed, where  $f$  is the tabular stress-strain function. (When  $\epsilon^* = 0$ , define  $\sigma_1 = E_0 \epsilon_1$ , where  $E_0$  is the modulus of elasticity on the MAT1 card)

For stiffness matrix generation the new material properties are:

$$E_1 = \begin{cases} \frac{\sigma_2 - \sigma_1}{\epsilon_2 - \epsilon_1} , & \text{if } \epsilon_2 \neq \epsilon_1 \\ E^* , & \text{if } \epsilon_2 = \epsilon_1 \end{cases} ; \quad (84)$$

and

$$G_1 = \frac{E_1}{E_0} G_0 , \quad (85)$$

where  $E_0$  and  $G_0$  are elastic moduli obtained from the MAT1 bulk data card via subroutine MAT. Note that  $E_1$  is calculated in PSBAR only to predict the next value of  $E$ , i.e., to update the ESTNL entry.

# MODULE FUNCTIONAL DESCRIPTIONS

For plastic element stresses and forces, the values are calculated in a fashion similar to that found in the phase 2 subroutine, SBAR2, (see section 4.87.2.5) of the SDR2 module.

They are:

$$\{\Delta P\} = [k_{aa}][T_{eb}]^T[C_a][E_a]\{\Delta u_a\} + [k_{ab}][T_{eb}]^T[C_b][E_b]\{\Delta u_b\}, \quad (86)$$

$$F_x = A\sigma_1, \quad (87)$$

$$V_1 = -\Delta P_2 + V_1^*, \quad (88)$$

$$V_2 = -\Delta P_3 + V_2^*, \quad (89)$$

$$T = -\Delta P_4 + T^*, \quad (90)$$

$$M_{1a} = -\Delta P_6 + M_{1a}^*, \quad (91)$$

$$M_{2a} = \Delta P_5 + M_{2a}^*, \quad (92)$$

$$M_{1b} = M_{1a} - V_1\ell, \quad (93)$$

$$M_{2b} = M_{2a} - V_2\ell. \quad (94)$$

The stresses due to bending, the axial stress, the minimum and maximum stresses, and the margins of safety are computed as in Equations 58 through 70.

The new ESTNL and ECPTNL entries are:

$$\epsilon_0^* = \epsilon^*, \quad (95)$$

$$\epsilon^* = \epsilon_1, \quad (96)$$

$$E^* = E_1, \quad (97)$$



# STRUCTURAL ELEMENT DESCRIPTIONS

$$V_1^* = V_1 , \quad (98)$$

$$V_2^* = V_2 , \quad (99)$$

$$T^* = T , \quad (100)$$

$$M_{1a}^* = M_{1a} , \quad (101)$$

$$M_{2a}^* = M_{2a} . \quad (102)$$

# MODULE FUNCTIONAL DESCRIPTIONS

## 4.87.2.8 "Consistent" Mass Matrix Calculation (Subroutine MCBAR of Module SMA2)

1. Generate the 12 by 12 matrix:

$$[M^e] = \frac{m}{420} \begin{bmatrix} 175 & 0 & 0 & 0 & 0 & 0 & 35 & 0 & 0 & 0 & 0 & 0 \\ & 156 & 0 & 0 & 0 & 22\ell & 0 & 54 & 0 & 0 & 0 & -13\ell \\ & & 156 & 0 & -22\ell & 0 & 0 & 0 & 54 & 0 & 13\ell & 0 \\ & & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & 4\ell^2 & 0 & 0 & 0 & -13\ell & 0 & -3\ell^2 & 0 \\ & & & & & 4\ell^2 & 0 & 13\ell & 0 & 0 & 0 & -3\ell^2 \\ & & & & & & 175 & 0 & 0 & 0 & 0 & 0 \\ & & & & & & & 156 & 0 & 0 & 0 & -22\ell \\ & & & & & & & & 156 & 0 & 22\ell & 0 \\ & & & & & & & & & 0 & 0 & 0 \\ & & & & & & & & & & 4\ell^2 & 0 \\ & & & & & & & & & & & 4\ell^2 \end{bmatrix}, \quad (103)$$

where

$$m = (\rho A + \mu)\ell \quad (104)$$

2. If "pin joints" exist ( $P_a$  or  $P_b$  nonzero), generate the "unpinned" 12 by 12 stiffness matrix in element coordinates,  $[K^e]$ , as in Equation 29, section 4.87.2.2.

For each pin joint of index  $j$ , perform the operations for  $i = 1, \dots, 12$  and  $\ell = 1, \dots, 12$ :

$$M_{i\ell}^p = M_{i\ell}^e - \frac{K_{\ell j}^e M_{ji}^e}{K_{jj}^e} - \frac{K_{ji}^e M_{\ell j}^e}{K_{jj}^e} + \frac{K_{\ell j}^e K_{ji}^e M_{jj}^e}{(K_{jj}^e)^2} \quad (105)$$

After each pin joint  $j$  operation, replace  $[M^e]$  by the "pinned" matrix  $[M^p]$ .

4. Partition the matrix into 6 by 6 submatrices:

# STRUCTURAL ELEMENT DESCRIPTIONS

$$[M] = \begin{bmatrix} M_{aa} & M_{ab} \\ M_{ba} & M_{bb} \end{bmatrix} \quad (106)$$

5. The matrices are converted to global coordinates by the equation:

$$[M_{ij}^g] = [T_{eb}]^T [C_i] [E_i]^T [M_{ij}] [T_{eb}]^T [C_j] [E_j] \quad (107)$$

where  $i$  is the pivot point (a or b) and  $j$  is used twice (for both a and b).

## 4.87.2.9 Thermal Analysis Calculations for the BAR Element

If a "stiffness" matrix for heat transfer analysis is to be generated, word 56 in COMMON data block SYSTEM is +1. The length,  $\ell$ , of the element is calculated with the structure analysis code, described in Section 4.87.2.2. The thermal conductivity coefficient,  $k$ , is obtained by calling subroutine HMAT, rather than MAT. The matrix terms are:

For the pivot point  $i$ ,

$$K_{ii} = \frac{k A}{\ell}.$$

For  $j \neq i$ ,

$$K_{ij} = -\frac{k A}{\ell}.$$

The "mass" matrix for heat transfer analysis is generated in subroutine MBAR. The capacity coefficient  $C_p$  is determined by subroutine HMAT and the matrix terms placed in the BGG matrix are:

$$B_{ii} = \frac{C_p A \ell}{2}.$$

The "stress" recovery is performed by subroutines SDHTF1, SDHTFF, and SDHTF2 in module SDR2. In Phase 1, the value  $k$  is extracted with subroutine HMAT and the output is:

$$K_1 = k,$$

$$[C] = \frac{1}{\ell} \begin{bmatrix} -1 & 1 \end{bmatrix}.$$

In Phase 2, the gradient and flux are:

$$\Delta T = [C] \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix}$$

$$Q = -K_1 \Delta T$$



## MODULE FUNCTIONAL DESCRIPTIONS

### 4.87.3 The SHEAR Panel and TWIST Panel Elements

#### 4.87.3.1 Input Data for SHEAR and TWIST Panels

1. The ECPT/EST entries for shear (SHEAR) and twist (TWIST) panel elements are:

| <u>Symbol</u>   | <u>Description</u>   |
|---|--|
| $SIL_i, i=1,2,3,4$  | Scalar indices for the connected points  |
| $\left. \begin{matrix} N_i, X_i, Y_i, Z_i \\ i = 1,2,3,4 \end{matrix} \right\}$ | Local coordinate system number and basic coordinate location for each of the connected points. |
| Mat I.D.  | Material identification number   |
| $t$   | Panel thickness  |
| $\mu$   | Nonstructural mass per unit area   |
| $t_\mu$   | Temperature for material properties  |

2. Coordinate system data

Using  $N_i, X_i, Y_i, Z_i, i = 1,2,3,4$  the program constructs  $[T_i], i = 1,2,3,4$ , the 3 by 3 global-to-basic transformation matrix for each point.

3. Material data

MAT I.D. and  $t_\mu$  are used, by utility routine MAT, to produce the following terms from the MPT and DIT data blocks:

| <u>Symbol</u>                  | <u>Description</u>             |
|--------------------------------|--------------------------------|
| $E$                            | Modulus of elasticity          |
| $G$                            | Shear modulus                  |
| $\nu$                          | Poisson's ratio                |
| $\rho$                         | Density                        |
| $\alpha$                       | Thermal expansion coefficient  |
| $T_0$                          | Reference temperature          |
| $g_e$                          | Structural damping coefficient |
| $\sigma_t, \sigma_c, \sigma_b$ | Stress limits                  |

## STRUCTURAL ELEMENT DESCRIPTIONS

### 4.87.3.2 Definition of Element Geometry

A mean plane is defined as parallel to the two diagonal lines and halfway between them. The projections of the points at the corners of the element onto the plane and the normal to the plane define the element coordinate system. Using standard vector algebra, the steps are:

1. Define:

$$\{V_{01}\} = \begin{Bmatrix} X_1 \\ Y_1 \\ Z_1 \end{Bmatrix}, \quad \{V_{02}\} = \begin{Bmatrix} X_2 \\ Y_2 \\ Z_2 \end{Bmatrix}, \text{ etc.} \quad (1)$$

2. Define diagonal vectors:

$$\{v_{d1}\} = \{V_{03}\} - \{V_{01}\}, \quad (2)$$

$$\{v_{d2}\} = \{V_{04}\} - \{V_{02}\}. \quad (3)$$

3. Define normal vector (x denotes cross product):

$$\{k_n\} = \{v_{d1}\} \times \{v_{d2}\}, \quad (4)$$

$$\{k\} = \frac{\{k_n\}}{|\{k_n\}|}, \quad (5)$$

$$A = \frac{1}{2} |\{k_n\}| \quad (\text{the projected area of the element}). \quad (6)$$

4. Define the vectors along the side of the element (see Figures 2 and 3)

$$\{v_{12}\} = \{V_{02}\} - \{V_{01}\}, \quad (7)$$

$$\{v_{41}\} = \{V_{01}\} - \{V_{04}\}. \quad (8)$$

# MODULE FUNCTIONAL DESCRIPTIONS

5. Define transformation matrix  $[T_e]$ , which transforms element coordinate to basic coordinates, using unit vectors:

$$\{v_{12}^p\} = \{v_{12}\} - (\{v_{12}\}^T \{k\}) \{k\} , \quad (9)$$

$$\{i\} = \frac{\{v_{12}^p\}}{|\{v_{12}^p\}|} , \quad (10)$$

$$\{j\} = \{k\} \times \{i\} , \quad (11)$$

$$[T_e] = \begin{bmatrix} i_1 & j_1 \\ i_2 & j_2 \\ i_3 & j_3 \end{bmatrix} . \quad (12)$$

6. Transform the four corner point of the element from basic coordinates to the element system:

$$\{r_1\} = \begin{Bmatrix} x_1 \\ y_1 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} , \quad (13)$$

$$\{r_2\} = \begin{Bmatrix} x_2 \\ y_2 \end{Bmatrix} = \begin{Bmatrix} |v_{12}^p| \\ 0 \end{Bmatrix} , \quad (14)$$

$$\{r_3\} = \begin{Bmatrix} x_3 \\ y_3 \end{Bmatrix} = [T_e]^T \{v_{d1}\} , \quad (15)$$

$$\{r_4\} = \begin{Bmatrix} x_4 \\ y_4 \end{Bmatrix} = - [T_e]^T \{v_{d1}\} . \quad (16)$$

The four corners of the element are now projected onto the mean plane.



# STRUCTURAL ELEMENT DESCRIPTIONS

7. The following conditions should be met. Otherwise, the interior angle at the indicated point is not valid.

$$y_3 > 0 \quad (\text{If not, the interior angle at point 2} > 180^\circ) \quad , (17)$$

$$x_3 > \frac{y_3}{y_4} x_4 \quad (\text{If not, the interior angle at point 4} > 180^\circ) \quad , (18)$$

$$y_4 > 0 \quad (\text{If not, the interior angle at point 1} > 180^\circ) \quad , (19)$$

$$x_4 < x_2 - (x_2 - x_3) \frac{y_4}{y_3} \quad (\text{If not, the interior angle at point 3} > 180^\circ) \quad . (20)$$

## 4.87.3.3 Coefficient Generation

The shape of the panel may be a parallelogram, a trapezoid, or a general quadrilateral, and the equations will be different for each case. The slopes of the opposite sides are checked for parallel effects, and the correct routine is used for each possibility.

1. Check for parallel effects:

If

$$\left| \frac{y_3 - y_4}{x_3 - x_4} \right| < \epsilon, \quad (21)$$

sides 1 and 3 of the panel are parallel ( $\epsilon = 10^{-1}$ ).

If

$$\left| \frac{y_4(x_3 - x_2) - y_3x_4}{x_4(x_3 - x_2) + y_4y_3} \right| < \epsilon, \quad (22)$$

sides 2 and 4 are parallel. If both terms are less than  $\epsilon$ , (i.e., the panel is a parallelogram), go to step (4). If both terms are greater than  $\epsilon$ , go to step (5). If the one pair of parallel sides is 1 and 3, go to step (2); if the one pair of parallel sides is 2 and 4, go to step (3).

# MODULE FUNCTIONAL DESCRIPTIONS

2. In this case the line connecting points 3 and 4 is approximately parallel to the line connecting points 1 and 2. The equations are:

$$y_p = \frac{x_2 y_3 y_4}{y_3 x_4 - y_4 (x_3 - x_2)} , \quad (23)$$

$$p_i = y_p - y_i \quad (i = 1, 2, 3, 4) , \quad (24)$$

$$x_p = \frac{x_2 y_3 x_4}{y_3 x_4 - y_4 (x_3 - x_2)} , \quad (25)$$

$$a = \left( \frac{x_2 - x_p}{y_p} \right) , \quad (26)$$

$$c = \left( \frac{x_1 - x_p}{y_p} \right) , \quad (27)$$

$$Z = \frac{p_1 p_2}{p_3 p_4} \frac{A}{2Gt} \left\{ 1 + \frac{2}{3(1+\nu)} (a^2 + ac + c^2) \right\} . \quad (28)$$

3. In this case the line connecting points 1 and 4 is approximately parallel to the line connecting points 2 and 3. The equations are:

$$d = -\frac{1}{2} \left[ \frac{x_4}{y_4} + \frac{x_3 - x_2}{y_3} \right] , \quad (29)$$

$$x_q = x_4 - \frac{x_3 - x_4}{y_3 - y_4} y_4 , \quad (30)$$

# STRUCTURAL ELEMENT DESCRIPTIONS

$$p_i = [(x_q - x_i) - y_i d] \frac{1}{\sqrt{1 + d^2}} \quad (i = 1, 2, 3, 4) \quad , \quad (31)$$

$$b = \frac{(x_q - x_4)d + y_4}{(x_q - x_4) - y_4 d} \quad , \quad (32)$$

$$Z = \frac{p_1 p_2}{p_3 p_4} \frac{A}{2Gt} \left\{ 1 + \frac{2}{3(1+\nu)} (b^2 + bd + d^2) \right\}. \quad (33)$$

4. In this case the panel approximates a parallelogram. The equations to solve are:

$$p_i = 1, \quad (i = 1, 2, 3, 4), \quad (34)$$

$$d = -\frac{1}{2} \left( \frac{x_4}{y_4} + \frac{x_3 - x_2}{y_3} + \frac{y_3 - y_4}{x_3 - x_4} \right) \quad , \quad (35)$$

$$Z = \frac{A}{2Gt} \left( 1 + \frac{2d^2}{1+\nu} \right). \quad (36)$$

5. In this case no parallel effects exist. The equations are:

$$x_q = x_4 - \frac{(x_3 - x_4)}{(y_3 - y_4)} y_4 \quad , \quad (37)$$



# MODULE FUNCTIONAL DESCRIPTIONS

$$x_p = \frac{x_2 x_4 y_3}{y_3 x_4 - y_4 (x_3 - x_2)} , \quad (38)$$

$$y_p = \frac{x_2 y_3 y_4}{y_3 x_4 - y_4 (x_3 - x_2)} , \quad (39)$$

$$\ell = \sqrt{(x_q - x_p)^2 + y_p^2} , \quad (40)$$

$$d = \frac{x_q - x_p}{y_p} , \quad (41)$$

$$p_i = \frac{y_p}{\ell} [(x_q - x_i) - y_i d] \quad (i = 1, 2, 3, 4) , \quad (42)$$

$$c = \frac{\ell}{p_1} - d , \quad (43)$$

$$b = \frac{\ell}{p_4} - c , \quad (44)$$

$$a = \frac{\ell}{p_2} - d . \quad (45)$$

# STRUCTURAL ELEMENT DESCRIPTIONS

Let:

$$\begin{aligned}
 F = & \frac{p_1 p_2 p_3 p_4}{2\ell^2} \{ [(a+b) + \frac{2}{3} (a^3+b^3) + \frac{1}{5} (a^5+b^5)] \log_e |a+b| \\
 & + [(c+d) + \frac{2}{3} (c^3+d^3) + \frac{1}{5} (c^5+d^5)] \log_e |c+d| \\
 & - [(b+c) + \frac{2}{3} (b^3+c^3) + \frac{1}{5} (b^5+c^5)] \log_e |b+c| \\
 & - [(d+a) + \frac{2}{3} (d^3+a^3) + \frac{1}{5} (d^5+a^5)] \log_e |d+a| \\
 & + \frac{1}{10} [(a^2-c^2) (b^3-d^3) + (b^2-d^2) (a^3-c^3)] \\
 & - \frac{1}{5} [(a-c) (b^4-d^4) + (b-d) (a^4-c^4)] \} .
 \end{aligned} \tag{46}$$

Then:

$$Z = \frac{p_1 p_2}{p_3 p_4} \frac{1}{2G\ell} \left\{ A + \frac{4}{1+\nu} \left[ F - \frac{2}{3} A \right] \right\} . \tag{47}$$

# MODULE FUNCTIONAL DESCRIPTIONS

## 4.87.3.4 Stiffness Matrix Formulation For a SHEAR Panel (Subroutine KPANEL of Module SMA1)

1. Calculate the lengths of the diagonals:

$$l_{13} = \sqrt{x_3^2 + y_3^2} , \quad (48)$$

$$l_{24} = \sqrt{(x_4 - x_2)^2 + y_4^2} , \quad (49)$$

2. Calculate the unit vectors along the diagonals:

$$u_1 = u_3 = \frac{x_3}{l_{13}} , \quad (50)$$

$$v_1 = v_3 = \frac{y_3}{l_{13}} , \quad (51)$$

$$u_2 = u_4 = \frac{x_4 - x_2}{l_{24}} , \quad (52)$$

$$v_2 = v_4 = \frac{y_4}{l_{24}} . \quad (53)$$

3. The loads along the diagonals in terms of the average shear stress along side 1 are:

$$A_1 = - \frac{x_2 y_4 l_{13}}{2(x_4 y_3 - x_3 y_4)} , \quad (54)$$



# STRUCTURAL ELEMENT DESCRIPTIONS

$$A_2 = \frac{x_2 y_3^2}{2(x_4 y_3 - x_3 y_4 - x_2(y_3 - y_4))} , \quad (55)$$

$$A_3 = -A_1 , \quad (56)$$

$$A_4 = -A_2 . \quad (57)$$

4. The loads at grid point  $i$  in terms of the displacements at grid point  $j$  may be expressed in terms of a (3x3) matrix  $[k_{ij}]$  where

$$[k_{ij}] = \frac{A_i A_j}{2Z} [T_i]^T [T_e] \begin{Bmatrix} u_i \\ v_i \end{Bmatrix} \{u_j v_j\}^T [T_e]^T [T_j] , \quad (58)$$

$$i = 1, 2, 3, 4 ,$$

$$j = 1, 2, \dots i .$$

5. The 3x3 matrices are related only to deflections and forces. The terms in the 6x6 matrices,  $[K_{ij}]$ , corresponding to rotations are zero. Expand the matrices to 6x6:

$$[K_{ij}] = \begin{bmatrix} k_{ij} & | & 0 \\ \hline 0 & | & 0 \end{bmatrix} . \quad (59)$$

The element structural damping matrix is equal to  $g_e$ , the structural damping coefficient, multiplied by the stiffness matrix,  $[K_{ij}]$ .

## 4.87.3.5 TWIST Element Stiffness Matrix Generation (Subroutine KPANEL of Module SMA1)

The following data for the SHEAR panel element are used for generation of the element stiffness matrix for the TWIST panel.

# MODULE FUNCTIONAL DESCRIPTIONS

|  |   |
|--|---|
| $[T_e]$  | The 3x2 transformation matrix (Equation 12).                                  |
| $x_2, x_3, x_4, y_3, y_4$  | The locations of the corners in the element system (Equations 14, 15 and 16). |
| $Z$  | The energy coefficient (Equation 28, 33, 36, or 47).                          |
| $\left. \begin{matrix} u_1 u_2 u_3 u_4 \\ v_1 v_2 v_3 v_4 \end{matrix} \right\}$ | Unit vector coefficients at the corners (Equations 50 through 53)             |
| $A_1, A_2, A_3, A_4$   | Load coefficients for the corners (Equations 54 through 57)                   |
| $[T_1], [T_2], [T_3], [T_4]$   | 3x3 global-to-basic transformation matrices                                   |

1. Generate the three by three matrices relating the moments at point i to the rotations at point j:

$$[q_{ij}] = \frac{A_i A_j t^2}{24Z} [T_i]^T [T_e] \begin{Bmatrix} -v_i \\ u_i \end{Bmatrix} \{-v_j u_j\}^T [T_e]^T [T_j] . \quad (60)$$

These are generated only for one point i (the pivot point) and  $j = 1, 2, 3$  and 4.

2. The 3x3 matrices  $[q_{ij}]$  are expanded to 6x6 matrices  $[K_{ij}]$  having zeros in the translational displacement rows and columns.

$$[K_{ij}] = \begin{bmatrix} 0 & r & 0 \\ - & | & - \\ 0 & | & q_{ij} \end{bmatrix} . \quad (61)$$

## 4.87.3.6 Mass Matrix Generation (Subroutine MASSTQ of Module SMA2)

The mass at each point is determined by cutting the quadrilateral into four overlapping triangles. Each triangle is defined by three of the four points as follows:

# STRUCTURAL ELEMENT DESCRIPTIONS

| Triangle No. | Connected Points |           |           |
|--------------|------------------|-----------|-----------|
| <u>K</u>     | <u>j1</u>        | <u>j2</u> | <u>j3</u> |
| I            | 4                | - 1       | - 2       |
| II           | 1                | - 2       | - 3       |
| III          | 2                | - 3       | - 4       |
| IV           | 3                | - 4       | - 1       |

The area of each triangle is determined by the equation :

$$A_K = \frac{1}{2} |(\{V_{oj2}\} - \{V_{oj1}\}) \times (\{V_{oj3}\} - \{V_{oj1}\})| , \quad (62)$$

where  $\{V_{oj1}\}$  is the location vector of the first point defining the triangle,  $\{V_{oj2}\}$  the second point, and  $\{V_{oj3}\}$  the third point.

The mass of each triangle is divided equally among its connected points. The mass at each point is :

$$m_1 = \frac{(\mu + \rho t)}{3} (A_4 + A_1 + A_2) , \quad (63)$$

$$m_2 = \frac{(\mu + \rho t)}{3} (A_1 + A_2 + A_3) , \quad (64)$$

$$m_3 = \frac{(\mu + \rho t)}{3} (A_2 + A_3 + A_4) , \quad (65)$$

$$m_4 = \frac{(\mu + \rho t)}{3} (A_3 + A_4 + A_1) . \quad (66)$$



## MODULE FUNCTIONAL DESCRIPTIONS

For each point a six by six diagonal mass matrix is constructed. The matrix is :

$$[M_{ii}] = \begin{bmatrix} m_i & & & & & \\ & m_i & & & & \\ & & m_i & & & \\ & & & 0 & & \\ & & & & 0 & \\ & & & & & 0 \end{bmatrix} \quad (67)$$

### 4.87.3.7 SHEAR Element Stress and Force Calculations (Subroutines SPANL1 and SPANL2 of Module SDR2)

The stress and force calculations are performed in two phases: phase 1 in SPANL1; phase 2 in SPANL2.

#### PHASE 1

1. Calculate the 1 by 3 matrices  $[S_i]$ ,  $i = 1, 2, 3, 4$ :

$$[S_i] = -\frac{A_i}{2Zt} \{u_i \ ; \ v_i\} [T_e]^T [T_i], \quad (68)$$

where  $A_i$ ,  $Z$ ,  $t$ ,  $u_i$ ,  $v_i$ ,  $[T_e]$  and  $[T_i]$  are as given in sections 4.87.3.2 and 4.87.3.3.

2. The  $[S]$  terms and the following parameters:

$$A_1, A_2, t, \frac{p_2}{p_1}, \frac{p_1 p_2}{p_3^2}, \frac{p_1 p_2}{p_4^2},$$

are saved on a scratch file for phase 2 calculations.  $p_i$ , where  $i = 1, 2, 3$  and  $4$  are calculated using the equations in section 4.87.3.3.

# STRUCTURAL ELEMENT DESCRIPTIONS

## PHASE 2

1. The average stress along side 1 is:

$$\bar{s}_1 = \sum_{i=1}^4 [S_i] \{u_i^t\}. \quad (69)$$

$\{u_i^t\}$  are the translational vectors where:

$$\{u_{gi}\} \Rightarrow \begin{Bmatrix} u_i^t \\ \text{---} \\ u_i^r \end{Bmatrix}. \quad (70)$$

2. The stresses on the corners are :

$$\tau_1 = \frac{p_2}{p_1} \bar{s}_1, \quad (71)$$

$$\tau_2 = \frac{p_1}{p_2} \bar{s}_1, \quad (72)$$

$$\tau_3 = \frac{p_1 p_2}{p_3^2} \bar{s}_1, \quad (73)$$

$$\tau_4 = \frac{p_1 p_2}{p_4^2} \bar{s}_1. \quad (74)$$

3. The average and maximum stresses are defined as :

$$\tau_{avg} = \frac{1}{4} (\tau_1 + \tau_2 + \tau_3 + \tau_4), \quad (75)$$

$$\tau_{max} = \max (|\tau_1|, |\tau_2|, |\tau_3|, |\tau_4|). \quad (76)$$

## MODULE FUNCTIONAL DESCRIPTIONS

4. The margin of safety in shear is defined by :

$$M.S._s = \begin{cases} \frac{\sigma_s}{\tau_{\max}} - 1, & \text{if } \sigma_s > 0 \\ \text{Integer "1",} & \text{if } \sigma_s \leq 0 \text{ or } \tau_{\max} = 0 \end{cases} \quad (77)$$

5. The net loads on the corners in the diagonal direction are :

$$P_{13} = A_1 \bar{s}_1 t, \quad (78)$$

$$P_{24} = A_2 \bar{s}_1 t. \quad (79)$$

### 4.87.3.8 TWIST Element Stress and Force Calculations (Subroutines SPANL1 and SPANL2 of Module SDR2)

The stress and force calculations are performed in two phases, as for the SHEAR panel element.

#### PHASE 1

1. Calculate for  $i = 1, 2, 3, 4$

$$[S_i] = -\frac{A_i}{4Z} \{-v_i; u_i\} [T_e]^T [T_i]. \quad (80)$$

2. The  $[S_i]$  terms and the following data:

$$A_1, A_2, t, \frac{p_2}{p_1}, \frac{p_1 p_2}{p_3^2}, \frac{p_1 p_2}{p_4^2}.$$

are saved on a scratch file for phase 2 calculations.

#### PHASE 2

1. The mean outer fibre shear stress along side 1 is:

$$\tau_1 = \sum_{i=1}^4 [S_i] \{u_i^r\}, \quad (81)$$



# STRUCTURAL ELEMENT DESCRIPTIONS

where  $\{u_i^r\}$  are the three rotational displacements:

$$\{u_{gi}\} \Rightarrow \begin{Bmatrix} u_i^t \\ u_i^r \end{Bmatrix} .$$

2. The stresses are :

$$\sigma_1 = \frac{p_2}{p_1} \tau_1, \quad (82)$$

$$\sigma_2 = \frac{\tau_1 p_1}{p_2}, \quad (83)$$

$$\sigma_3 = \frac{p_1 p_2}{p_3^2} \tau_1, \quad (84)$$

$$\sigma_4 = \frac{p_1 p_2}{p_4^2} \tau_1, \quad (85)$$

$$\sigma_{avg} = \frac{1}{4} (\sigma_1 + \sigma_2 + \sigma_3 + \sigma_4), \quad (86)$$

$$\sigma_{max} = \max(|\sigma_1|, |\sigma_2|, |\sigma_3|, |\sigma_4|). \quad (87)$$

3. The margin of safety in shear is defined by :

$$M.S._s = \begin{cases} \frac{\sigma_s}{|\sigma_{max}|} - 1, & \sigma_s > 0 \\ \text{Integer "1",} & \sigma_s \leq 0 \text{ or } \sigma_{max} = 0 \end{cases} \quad (88)$$

4. The moments are :

$$M_{13} = \frac{A_1 t^2}{6} \tau_1, \quad (89)$$

# MODULE FUNCTIONAL DESCRIPTIONS

$$M_{24} = \frac{A_2 t^2}{6} \tau_1 \quad (90)$$

## 4.87.3.9 SHEAR Panel Differential Stiffness Calculations (Subroutine DSHEAR of Module DSMG1)

### 1. Data

The data necessary for analysis are included in the ECPT, CSTM, MPT and UGV data blocks. The following data are generated as in sections 4.87.3.2, 4.87.3.3, and 4.87.3.4 .

- a.  $[T_i]$ ,  $i = 1,2,3,4$ , the 3x3 transformation matrices between global and basic coordinates, at the four corners of the shear panel.
- b.  $[T_e]$ , the 3x2 transformations between basic and element coordinates.
- c.  $\{k\}$ , the unit vector normal to the plane in basic coordinates.
- d.  $u_i, v_i$ ,  $i = 1,2,3,4$ , the unit vectors along the diagonals in element coordinates.
- e.  $A_i$ ,  $i = 1,2,3,4$ , the load coefficients for the corners.
- f.  $Z$ , the energy coefficient for the panel.
- g.  $\ell_{13}$  and  $\ell_{24}$ , the lengths of the diagonals.

### 2. Algorithm

- a. The load in the diagonal between points 1 and 3 is :

$$F_{13} = - \frac{A_1}{2Z} \sum_{i=1}^4 A_i \{u_i \quad v_i\}^T [T_e]^T [T_i] \begin{Bmatrix} x_{1i} \\ x_{2i} \\ x_{3i} \end{Bmatrix} \quad (91)$$

# STRUCTURAL ELEMENT DESCRIPTIONS

where  $\{x_i\}$  is the vector of the three translations in global coordinates for point (i).

b. The load in the other diagonal is :

$$F_{24} = \frac{A_2}{A_1} F_{13} . \quad (92)$$

c. Construct a perpendicular, which is defined in basic coordinates, to each diagonal vector in the plane of the panel.

$$\{j_1\} = [T_e] \begin{Bmatrix} -v_1 \\ u_1 \end{Bmatrix} , \quad (93)$$

$$\{j_2\} = [T_e] \begin{Bmatrix} -v_2 \\ u_2 \end{Bmatrix} . \quad (94)$$

d. The nonzero partitions of the overall differential stiffness matrix for the displacements are:

$$[k_{11}^d] = F'_{13} [T_1]^T [\{j_1\}\{j_1\}^T + \{k\}\{k\}^T] [T_1] , \quad (95)$$

$$[k_{13}^d] = -F'_{13} [T_1]^T [\{j_1\}\{j_1\}^T + \{k\}\{k\}^T] [T_3] , \quad (96)$$

$$[k_{33}^d] = F'_{13} [T_3]^T [\{j_1\}\{j_1\}^T + \{k\}\{k\}^T] [T_3] , \quad (97)$$

$$[k_{31}^d] = [k_{13}^d]^T , \quad (98)$$

$$[k_{22}^d] = F'_{24} [T_2]^T [\{j_2\}\{j_2\}^T + \{k\}\{k\}^T] [T_2] , \quad (99)$$

$$[k_{24}^d] = -F'_{24} [T_2]^T [\{j_2\}\{j_2\}^T + \{k\}\{k\}^T] [T_4] , \quad (100)$$



# MODULE FUNCTIONAL DESCRIPTIONS

$$[k_{44}^d] = F_{24}' [T_4]^T [\{j_2\}\{j_2\}^T + \{k\}\{k\}^T] [T_4], \quad (101)$$

$$[k_{42}^d] = [k_{24}^d]^T, \quad (102)$$

where :

$$F_{13}' = \frac{F_{13}}{\ell_{13}}, \quad (103)$$

$$F_{24}' = \frac{F_{24}}{\ell_{24}}. \quad (104)$$

5. The actual 6x6 partitions are

$$[K_{ij}^d] = \begin{bmatrix} k_{ij}^d & 0 \\ 0 & 0 \end{bmatrix}, \quad (105)$$

and

$$[K_{ji}^d] = [K_{ij}^d]^T. \quad (106)$$

# STRUCTURAL ELEMENT DESCRIPTIONS

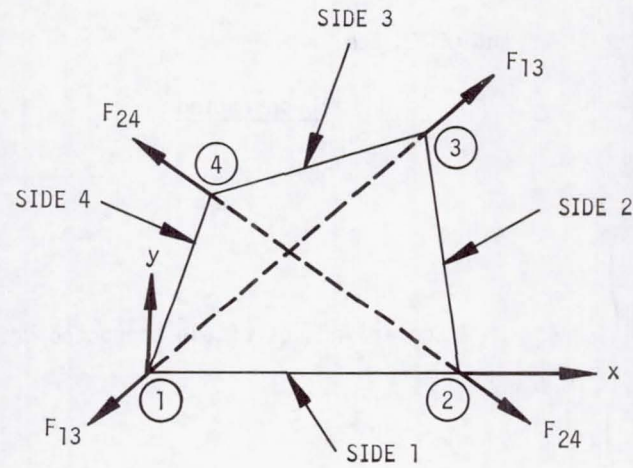


Figure 2. Shear panel element coordinate system and element forces.

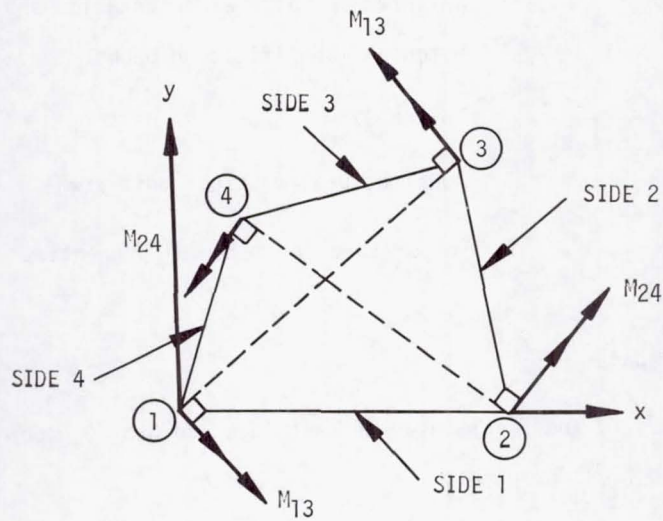


Figure 3. Twist panel element coordinate system and element forces.

# MODULE FUNCTIONAL DESCRIPTIONS

## 4.87.4 TRMEM and QDMEM Elements

### 4.87.4.1 Input Data for the TRMEM and QDMEM Elements

1. ECPT entries for the TRMEM and QDMEM are:

| <u>Symbol</u>    | <u>Description</u>  |  |
|------------------|---|--|
| <u>TRMEM</u>     | <u>QDMEM</u>  |  |
| SIL <sub>1</sub> | SIL <sub>1</sub>  |  |
| SIL <sub>2</sub> | SIL <sub>2</sub>  |  |
| SIL <sub>3</sub> | SIL <sub>3</sub>  |  |
|                  | SIL <sub>4</sub>  |  |
|                  | Scalar indices of the connected grid points.  |  |
| N <sub>i</sub>   | N <sub>i</sub>  |  |
| X <sub>i</sub>   | X <sub>i</sub>  |  |
| i = 1,3          | i = 1,4   |  |
| Y <sub>i</sub>   | Y <sub>i</sub>  |  |
| Z <sub>i</sub>   | Z <sub>i</sub>  |  |
|                  | Local coordinate system numbers and location coordinates in the basic system for the connected grid points. |  |
| θ                | Anisotropic material orientation angle  |  |
| Mat I.D.         | Material identification number  |  |
| t                | Thickness   |  |
| μ                | Nonstructural mass per unit area  |  |
| t <sub>μ</sub>   | Temperature for material properties   |  |

### 2. Coordinate system data

The numbers N<sub>i</sub>, X<sub>i</sub>, Y<sub>i</sub> and Z<sub>i</sub> are used to calculate the 3 by 3 global-to-basic coordinate transformation matrices [T<sub>i</sub>] for points i = 1, 2, 3, and 4.



## STRUCTURAL ELEMENT DESCRIPTIONS

### 3. Material data

| <u>Symbol</u>                     | <u>Description</u>                               |
|-----------------------------------|--|
| $[G_e]$                           | 3x3 stress-strain matrix                         |
| $\rho$                            | Mass density                                     |
| $\alpha_x, \alpha_y, \alpha_{xy}$ | Three thermal expansion coefficients             |
| $T_o$                             | Reference temperature                            |
| $g_e$                             | Structural damping coefficient                   |
| $\sigma_t, \sigma_c, \sigma_s$    | Stress limits for tension, compression and shear |

### 4.87.4.2 Basic Equations For TRMEM

1. The element coordinate system is defined by the following equations (see Figure 4)

$$\{V_{12}\} = \begin{Bmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{Bmatrix}, \quad (1)$$

$$\{V_{13}\} = \begin{Bmatrix} x_3 - x_1 \\ y_3 - y_1 \\ z_3 - z_1 \end{Bmatrix}, \quad (2)$$

$$\{i\} = \frac{\{V_{12}\}}{|\{V_{12}\}|}, \quad (3)$$

$$\{k\} = \frac{\{i\} \times \{V_{13}\}}{|\{i\} \times \{V_{13}\}|}, \quad (4)$$

$$\{j\} = \{k\} \times \{i\}. \quad (5)$$

2. The displacement transformation matrix from basic coordinates to in-plane coordinates is :

# MODULE FUNCTIONAL DESCRIPTIONS

$$[E]^T = \begin{bmatrix} i_1 & i_2 & i_3 \\ j_1 & j_2 & j_3 \end{bmatrix}, \quad (6)$$

3. The coordinates of the points in the element coordinate system are :

$$x_1 = y_1 = z_1 = 0, \quad (7)$$

$$x_2 = |\{V_{12}\}|, \quad (8)$$

$$x_3 = \{V_{13}\}^T \{i\}, \quad (9)$$

$$y_3 = |\{i\} \times \{V_{13}\}|. \quad (10)$$

The area is :

$$A = \frac{1}{2} x_2 y_3. \quad (11)$$

4. The transformations from displacements at the points to strains are:

$$[C_1] = \begin{bmatrix} -\frac{1}{x_2} & 0 \\ 0 & \frac{1}{y_3} \left( \frac{x_3}{x_2} - 1 \right) \\ \frac{1}{y_3} \left( \frac{x_3}{x_2} - 1 \right) & -\frac{1}{x_2} \end{bmatrix}, \quad (12)$$

# STRUCTURAL ELEMENT DESCRIPTIONS

$$[C_2] = \begin{bmatrix} \frac{1}{x_2} & 0 \\ 0 & -\frac{x_3}{x_2 y_3} \\ -\frac{x_3}{x_2 y_3} & \frac{1}{x_2} \end{bmatrix}, \quad (13)$$

$$[C_3] = \begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{y_3} \\ \frac{1}{y_3} & 0 \end{bmatrix}. \quad (14)$$

## 4.87.4.3 Stiffness Matrix Calculation for TRMEM (Subroutine KTRMEM of Module SMA1)

1. The equation used in the stiffness matrix generation in global coordinates is:

$$[k_{ij}] = At([C_i][E]^T[T_i])^T[G_e]([C_j][E]^T[T_j]), \quad (15)$$

where "i" is the pivot point number, and j = 1, 2, 3 are the three connected points.  $[k_{ij}]$  is a 3x3 matrix.

2. For use in the overall structural matrix, the matrices are expanded to 6x6 to form:

$$[K_{ij}] = \begin{bmatrix} k_{ij} & 0 \\ 0 & 0 \end{bmatrix}. \quad (16)$$



# MODULE FUNCTIONAL DESCRIPTIONS

## 4.87.4.4 Mass Matrix Calculation for the TRMEM Element (Subroutine MASSTQ of Module SMA2)

The mass is generated by the following algorithm.

The vectors defining the sides are :

$$\{V_{12}\} = \begin{pmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{pmatrix}, \quad (17)$$

$$\{V_{13}\} = \begin{pmatrix} x_3 - x_1 \\ y_3 - y_1 \\ z_3 - z_1 \end{pmatrix}. \quad (18)$$

The area is :

$$A = \frac{1}{2} |\{V_{12}\} \times \{V_{13}\}|. \quad (19)$$

The mass at each point is :

$$m = \frac{A}{3} (\rho t + \mu), \quad (20)$$

which is one-third of the total mass.

For each point the diagonal mass matrix is :

$$[M_{ii}] = \begin{bmatrix} m & & & \\ & m & & \\ & & m & \\ & & & 0 \end{bmatrix}, \quad i = 1, 2, 3. \quad (21)$$

## STRUCTURAL ELEMENT DESCRIPTIONS

### 4.87.4.5 Element Load Calculations For The TRMEM Element (Subroutine TRIMEM of Module SSG1)

Using the loading temperature on the element,  $\bar{T}$ , given in the GPTT data block, the triangular membrane routine generates force vectors by the equation:

$$\{P_i\} = At[T_i]^T[E][C_i]^T[G_e]\{\alpha\}(\bar{T} - T_0), \quad i = 1, 2, 3 \quad (22)$$

where  $\{P_i\}$  is a 3x1 vector.

The forces are placed in the PG load vector data block.

### 4.87.4.6 Element Stress Calculations For The TRMEM Element (Subroutines STRME1 and STQME2 of Module SDR2)

#### 1. Calculations performed in STRME1 (Phase 1 calculations).

a. Using the formulae given in section 4.87.4.2, calculate the following terms:

|         |               |                |
|---------|---------------|----------------|
| $[C_i]$ | $i = a, b, c$ | $(3 \times 2)$ |
| $[T_i]$ | $i = a, b, c$ | $(3 \times 3)$ |
| $[E]$   |               | $(3 \times 2)$ |
| $[G_e]$ |               | $(3 \times 3)$ |

The transformations from displacements to stress are:

$$[S_i] = [G_e][C_i][E]^T[T_i] \quad (24)$$

Equation 23 is intentionally missing.

# MODULE FUNCTIONAL DESCRIPTIONS

The temperature to stress relation is:

$$\{S_t\} = - [G_e]\{\alpha\} \quad (25)$$

where

$$\{\alpha\} = \alpha \begin{Bmatrix} 1 \\ 1 \\ 0 \end{Bmatrix}, \quad (26)$$

for isotropic materials.  $\{\alpha\}$  is input by the user for anisotropic materials and corrected for material angle by  $\alpha = [v]\{\alpha_m\}$ .

## 2. Calculations performed by STQME2 (Phase 2 calculations)

The equation for stress is:

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{Bmatrix} = \left[ \sum_{i=a,b,c} [S_i]\{u_{gi}\} \right] + \{S_t\} [\bar{T} - T_0], \quad (27)$$

where  $\bar{T}$  is the loading temperature obtained from the GPTT data block.

The principal stresses are:

$$\sigma_1 = \left( \frac{\sigma_x + \sigma_y}{2} \right) + \sqrt{\left( \frac{\sigma_x - \sigma_y}{2} \right)^2 + \sigma_{xy}^2}, \quad (28)$$

$$\sigma_2 = \left( \frac{\sigma_x + \sigma_y}{2} \right) - \sqrt{\left( \frac{\sigma_x - \sigma_y}{2} \right)^2 + \sigma_{xy}^2}, \quad (29)$$

$$\theta = \frac{1}{2} \arctan \left( \frac{2\sigma_{xy}}{\sigma_x - \sigma_y} \right) \quad (\text{in degrees}), \quad (30)$$

where  $\theta$  is limited to:  $-90^\circ \leq \theta \leq 90^\circ$



## STRUCTURAL ELEMENT DESCRIPTIONS

The maximum shear is :

$$\tau = \sqrt{\left(\frac{\sigma_x - \sigma_y}{2}\right)^2 + \sigma_{xy}^2} \quad (31)$$

### 4.87.4.7 Differential Stiffness Matrix Calculations for the TRMEM Element (Subroutine DTRMEM of Module DSMG1)

#### 1. Input Data.

ECPT for element

i - Pivot point scalar index

{u<sub>1</sub>}, {u<sub>2</sub>}, {u<sub>3</sub>} - Displacements of pivots on triangle (UGV)

$\bar{T}$  - Average loading temperature of the grid points of the element (GPTT)

CSTM - coordinate systems

MPT - Material properties table

#### 2. Output Data.

[K<sub>i1</sub><sup>d</sup>], [K<sub>i2</sub><sup>d</sup>], [K<sub>i3</sub><sup>d</sup>] - partitions of the differential stiffness matrix.

#### 3. Solution Algorithm.

a. The planar stresses in the element,  $\sigma_x$ ,  $\sigma_y$  and  $\sigma_{xy}$ , are calculated as in the SDR2 (Stress Data Recovery) module. The following data are saved for use in the differential stiffness calculation :

{i}, {j}, {k} - Unit vectors defining the element coordinate system.

A, t - Area and thickness

x<sub>2</sub>, x<sub>3</sub>, y<sub>3</sub> - Locations of the points, element coordinates

[T<sub>1</sub>], [T<sub>2</sub>], [T<sub>3</sub>] - Global-to-basic coordinate transformations

$\sigma_x$ ,  $\sigma_y$ ,  $\sigma_{xy}$  - Stresses in element system

# MODULE FUNCTIONAL DESCRIPTIONS

b. The differential stiffness matrix in terms of the six generalized coordinates

( $\omega_x$ ,  $\omega_y$ ,  $\omega_z$ ,  $\epsilon_{xx}$ ,  $\epsilon_{yy}$ , and  $\epsilon_{xy}$ ) is:

$$[K_g^d] = At \begin{bmatrix} \sigma_y & -\sigma_{xy} & 0 & 0 & 0 & 0 \\ -\sigma_{xy} & \sigma_x & 0 & 0 & 0 & 0 \\ 0 & 0 & (\sigma_x + \sigma_y) & -\sigma_{xy} & \sigma_{xy} & (\sigma_x - \sigma_y) \\ 0 & 0 & -\sigma_{xy} & 0 & 0 & 0 \\ 0 & 0 & \sigma_{xy} & 0 & 0 & 0 \\ 0 & 0 & (\sigma_x - \sigma_y) & 0 & 0 & 0 \end{bmatrix} \quad (32)$$

If the subroutine is called from the DTRIA or DQUAD routines the following terms are set to zero:

$$[K_{g11}^d] = [K_{g12}^d] = [K_{g21}^d] = [K_{g22}^d] = 0$$

c. The transformation matrices from displacements at the points to generalized coordinates are:

$$[C_1^d] = \begin{bmatrix} 0 & 0 & \gamma_3 - \gamma_2 \\ 0 & 0 & \gamma_1 \\ \frac{\gamma_2 - \gamma_3}{2} & -\frac{\gamma_1}{2} & 0 \\ -\gamma_1 & 0 & 0 \\ 0 & \gamma_3 - \gamma_2 & 0 \\ \frac{\gamma_2 - \gamma_3}{2} & -\frac{\gamma_1}{2} & 0 \end{bmatrix} \quad (33)$$

# MODULE FUNCTIONAL DESCRIPTIONS

$$[c_2^d] = \begin{bmatrix} 0 & 0 & -\gamma_3 \\ 0 & 0 & -\gamma_1 \\ \frac{\gamma_3}{2} & \frac{\gamma_1}{2} & 0 \\ \gamma_1 & 0 & 0 \\ 0 & -\gamma_3 & 0 \\ \frac{\gamma_3}{2} & \frac{\gamma_1}{2} & 0 \end{bmatrix} \quad (34)$$

$$[c_3^d] = \begin{bmatrix} 0 & 0 & \gamma_2 \\ 0 & 0 & 0 \\ -\frac{\gamma_2}{2} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \gamma_2 & 0 \\ \frac{\gamma_2}{2} & 0 & 0 \end{bmatrix} \quad (35)$$



# STRUCTURAL ELEMENT DESCRIPTIONS

where

$$\gamma_1 = \frac{1}{x_2} , \quad (36)$$

$$\gamma_2 = \frac{1}{y_3} , \quad (37)$$

$$\gamma_3 = \frac{x_3}{x_2 y_3} . \quad (38)$$

d. The partitions (3x3) of the differential stiffness matrix in global coordinates are :

$$[K_{ij}] = ([C_i^d] [E^d]^T [T_i])^T [K_\omega^d] [C_j^d] [E^d]^T [T_j] , \quad (39)$$

i = pivot point

j = 1, 2 and 3

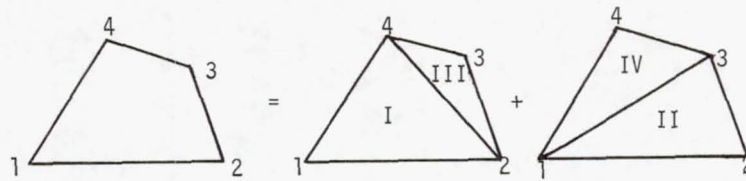
where

$$[E^d]^T = \begin{bmatrix} \{i\}^T \\ \{j\}^T \\ \{k\}^T \end{bmatrix} \quad (3 \times 3) . \quad (40)$$

4.87.4.8 General Calculations for the QDMEM by the QDMEM Driver Routines (Subroutines KQDMEM of Module SMA1, SQDMEM of Module SDR2, DQDMEM of Module DSMG1).

1. The quadrilateral is divided into four triangles as shown in the figure below:

# MODULE FUNCTIONAL DESCRIPTIONS



The thickness used for each triangle is one-half that given for the quadrilateral. Since no special calculation time is saved by generating a unique element coordinate system, the basic locations of the points are used to calculate individual coordinate systems for the triangles.

An integer mapping matrix  $[M]$  containing the quadrilateral point numbers is used to convert point numbers for the triangles to point numbers for the quadrilateral.

$$[M] = \begin{array}{c} \begin{array}{ccc} \text{Triangle Point No.} & & \text{Triangle No.} \\ \begin{bmatrix} \underline{a} & \underline{b} & \underline{c} \\ 1 & 2 & 4 \\ 2 & 3 & 1 \\ 3 & 4 & 2 \\ 4 & 1 & 3 \end{bmatrix} & & \begin{array}{l} \text{(I)} \\ \text{(II)} \\ \text{(III)} \\ \text{(IV)} \end{array} \end{array} \end{array} \quad (41)$$

The data corresponding to the point numbers in each row of the matrix are transferred to the triangular membrane routine. The pivot grid point  $i$  is also transferred.

## 2. Material orientation for subtriangles.

The material orientation angle for the QDMEM element must be transformed to a set of angles related to the base of each subtriangle. This requires the following steps:

# STRUCTURAL ELEMENT DESCRIPTIONS

1. The element coordinate system is defined as follows:

$$\{V_i\} = \begin{Bmatrix} X_i \\ Y_i \\ Z_i \end{Bmatrix} \quad i = 1, 2, 3, 4, \quad (42)$$

$$\{d_{21}\} = \{V_2\} - \{V_1\} \quad , \quad (43)$$

$$\{i\} = \frac{\{d_{21}\}}{|\{d_{21}\}|} \quad , \quad (44)$$

$$\{d_{41}\} = \{V_4\} - \{V_1\} \quad , \quad (45)$$

$$\{k\} = \frac{\{i\} \times \{d_{41}\}}{|\{i\} \times \{d_{41}\}|} \quad , \quad (46)$$

$$\{j\} = \{k\} \times \{i\} \quad . \quad (47)$$

The material is oriented for each triangle as follows:

$$s_1 = \sin(\theta) \quad , \quad (48)$$

$$c_1 = \cos(\theta) \quad , \quad (49)$$

$$\{p\} = c_1 \{i\} + s_1 \{j\} \quad , \quad (50)$$

$$\{V_{II}^t\} = \{V_3\} - \{V_2\} \quad , \quad (51)$$

$$\{V_{III}^t\} = \{V_4\} - \{V_3\} \quad , \quad (52)$$

$$\{V_{IV}^t\} = \{V_1\} - \{V_4\} \quad , \quad (53)$$



## MODULE FUNCTIONAL DESCRIPTIONS

$$c_i = \frac{\{V_i^t\}^T \{p\}}{|\{V_i^t\}|} = \cos (\theta_i) , \quad (54)$$

$i = \text{II, III and IV}$

$$s_i = \frac{(\{V_i^t\} \times \{p\})^T \{k\}}{|\{V_i^t\}|} = \sin (\theta_i) . \quad (55)$$

The values  $s_i$  and  $c_i$  may be passed to the triangular membrane subroutines in lieu of the angles  $\theta_i$ .

### 4.87.4.9 Stiffness Matrix Calculations for the QDMEM.

Three stiffness matrices, which the triangular membrane routine calculates for each sub-triangle, are added to the four matrices which will be output. (Note: only three triangles are needed for each pivot point.) For example, consider the case where point 2 is the pivot grid point. (i.e., the second SIL value in the grid point connection list equals the pivot grid point SIL value). Triangle I is calculated by entering the geometry and property data for the 1, 2 and 4 points on the quadrilateral, with number 2 as the pivot point. The outputs from the stiffness matrix generation routines for the TRMEM are:

$$[K_{21}], [K_{22}], [K_{24}]$$

Data for triangles II and III are also entered, and their corresponding matrix partitions are added. Triangle number IV is not connected to point 2.

### 4.87.4.10 Element Stress Calculations for the QDMEM (Subroutine SQDME1 and STQME2 of Module SDR2).

The solution for stress in the quadrilateral involves two phases. In the first phase (SQDME1) the triangular membrane partitions are solved for their stress-displacement matrices. These matrices are modified to correspond to the element coordinate system. They

## STRUCTURAL ELEMENT DESCRIPTIONS

are added together to form four 3x3 matrices relating displacements in global coordinates to element stress. A vector is also calculated which transforms temperature to stress.

The second phase (STQME2) involves the acquisition of the displacement and temperature data and the calculation of the net stress.

The following steps are used to set up an element coordinate system and obtain triangle to element stress transformations.

### Phase I

1. The following quantities are calculated:

$$\{V_i\} = \begin{Bmatrix} X_i \\ Y_i \\ Z_i \end{Bmatrix} \quad i = 1, 2, 3, 4, \quad (56)$$

$$\{d_1\} = \{V_3\} - \{V_1\}, \quad (57)$$

$$\{d_2\} = \{V_4\} - \{V_2\}, \quad (58)$$

$$\{k\} = \frac{\{d_1\} \times \{d_2\}}{|\{d_1\} \times \{d_2\}|}, \quad (59)$$

$$\{a_{12}\} = \{V_2\} - \{V_4\}, \quad (60)$$

$$\{a_{23}\} = \{V_3\} - \{V_2\}, \quad (61)$$

$$\{a_{41}\} = \{V_1\} - \{V_4\}, \quad (62)$$

$$\{a_{34}\} = \{V_4\} - \{V_3\}, \quad (63)$$

# MODULE FUNCTIONAL DESCRIPTIONS

$$h = \{a_{12}\}^T \{k\} . \quad (64)$$

$h$  is the perpendicular distance between the diagonals. The mean plane of the element lies halfway between the diagonals.

2. The unit vectors along the edges of the four triangles, projected on the mean plane, are calculated from:

$$x_2 = |\{a_{12}\} - h\{k\}| , \quad (65)$$

$$\{i\} = \frac{\{a_{12}\} - h\{k\}}{x_2} , \quad (66)$$

$$\{j\} = \{k\} \times \{i\} , \quad (67)$$

$$[R] = \begin{bmatrix} \{i\}^T \\ \{j\}^T \end{bmatrix} \quad (2 \times 3) , \quad (68)$$

$$\{v_{12}\} = \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} , \quad (69)$$

$$\{v_{ij}\} = [R]\{a_{ij}\} \quad ij = 23, 34, 41 , \quad (70)$$

$$\{w^I\} = \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} , \quad (71)$$

$$\{w^{II}\} = \frac{1}{|\{v_{23}\}|} \{v_{23}\} , \quad (72)$$

$$\{w^{III}\} = \frac{1}{|\{v_{34}\}|} \{v_{34}\} , \quad (73)$$

$$\{w^{IV}\} = \frac{1}{|\{v_{41}\}|} \{v_{41}\} , \quad (74)$$

3. For each triangular membrane,  $\beta = I, II, III, IV$ , of the quadrilateral, the subroutine STRMEL is called to calculate the three stress functions  $[S_i]$  where



# STRUCTURAL ELEMENT DESCRIPTIONS

$$\begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_{12} \end{pmatrix}_{\text{triangle}} = \sum_{i=1}^3 [S_i] \{u_{gdi}\}, \quad (75)$$

where  $\{u_{gdi}\}$  are the displacements in global coordinates of the points on the triangle.

$[S_i^\beta]$  is calculated using the full thickness of the panel for triangle  $\beta$ .

4. The stress functions, Equation 24, are transformed to the element coordinates by the matrix  $[T^\beta]$ :

$$[S_i^\beta]_{\text{element}} = [T^\beta][S_i^\beta], \quad (76)$$

where

$$[T^\beta] = \begin{bmatrix} w_1^2 & w_2^2 & -2w_1w_2 \\ w_2^2 & w_1^2 & 2w_1w_2 \\ w_1w_2 & -w_1w_2 & w_2^2 - w_1^2 \end{bmatrix}, \quad (77)$$

and

$$\begin{Bmatrix} w_1 \\ w_2 \end{Bmatrix} = \{w^\beta\} \quad (78)$$

for triangle  $\beta = I, II, III, IV$ .

5. Using the mapping matrix  $[M]$ , Equation 41, the matrices are added. The actual equations are:

$$[\bar{S}_1] = \frac{1}{4}([S_a^I] + [S_c^{II}] + [S_b^{IV}]) \quad (79)$$

$$[\bar{S}_2] = \frac{1}{4}([S_b^I] + [S_a^{II}] + [S_c^{III}]) \quad (80)$$

# MODULE FUNCTIONAL DESCRIPTIONS

$$[\bar{S}_3] = \frac{1}{4}([S_b^{II}] + [S_a^{III}] + [S_c^{IV}]) , \quad (81)$$

$$[\bar{S}_4] = \frac{1}{4}([S_c^I] + [S_b^{II}] + [S_a^{IV}]) , \quad (82)$$

$$\{\bar{S}_t\} = \frac{1}{4}(\{S_t^I\} + \{S_t^{II}\} + \{S_t^{III}\} + \{S_t^{IV}\}) . \quad (83)$$

where  $[S_i^\beta]$   $i = a, b, c$ , are the stress matrices in Equation 76 and  $\{S_t^\beta\}$  is the vector in Equation 25.

## Phase 2

1. In phase 2 the stresses are calculated from:

$$\begin{pmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{pmatrix} = \{\bar{S}_t\} \bar{t} + \sum_{i=1}^4 [\bar{S}_i] \{u_{qdi}\} , \quad (84)$$

where

$$\bar{t} = \frac{1}{4} \sum_{i=1}^4 (t_i - T_0) , \quad (85)$$

and  $t_i$  are the loading temperatures at the grid points, obtained from the GPTT data block.

2. The principal stresses and the angles are calculated in exactly the same manner as for the TRMEM element.

### 4.87.4.11 Mass Matrix Generation for the QDMEM Element (Subroutine MASSTQ of Module SMA2).

The mass is generated by the following algorithm.

The vectors defining the sides and the diagonals are:

$$\{V_{ij}\} = \begin{pmatrix} X_j - X_i \\ Y_j - Y_i \\ Z_j - Z_i \end{pmatrix} , \quad ij = 12, 23, 34, 41, 13, 24 \quad (86)$$

# STRUCTURAL ELEMENT DESCRIPTIONS

The area of the subtriangles defined by the integer mapping matrix [M], Equation 41, is:

$$A_I = \frac{1}{2} |\{V_{14}\} \times \{V_{12}\}| , \quad (87)$$

$$A_{II} = \frac{1}{2} |\{V_{12}\} \times \{V_{23}\}| , \quad (88)$$

$$A_{III} = \frac{1}{2} |\{V_{23}\} \times \{V_{34}\}| , \quad (89)$$

$$A_{IV} = \frac{1}{2} |\{V_{34}\} \times \{V_{14}\}| . \quad (90)$$

The area of the quadrilateral is:

$$A_q = \frac{1}{2} |\{V_{13}\} \times \{V_{24}\}| . \quad (91)$$

The mass at each point is:

$$m_1 = \frac{(A_I + A_q)}{3} (\mu + \rho t) , \quad (92)$$

$$m_2 = \frac{(A_{II} + A_q)}{3} (\mu + \rho t) , \quad (93)$$

$$m_3 = \frac{(A_{III} + A_q)}{3} (\mu + \rho t) , \quad (94)$$

$$m_4 = \frac{(A_{IV} + A_q)}{3} (\mu + \rho t) . \quad (95)$$

For each point, the diagonal mass matrix is:

$$[M_{ii}] = \begin{bmatrix} \overline{m_i} & & & \\ & m_i & & \\ & & \overline{m_i} & \\ & & & 0 \end{bmatrix} , \quad i = 1, 2, 3, 4 . \quad (96)$$



## MODULE FUNCTIONAL DESCRIPTIONS

### 4.87.4.12 Thermal Load Computation For The QDMEM.

The thermal loads are calculated using the triangular membrane routine. The EST data are rearranged to correspond to each of the four subtriangles, and each triangle produces a load in global coordinates.

### 4.87.4.13 Differential Stiffness Computations For The QDMEM (Subroutines DQDMEM and DTRMEM of Module DSMG1).

The differential stiffness matrices for the QDMEM element are generated by rearranging the ECPT data into four sets of TRMEM data. The TRMEM differential stiffness routine calculates the stresses, generates the differential stiffness matrix partitions in global coordinates and inserts them in the overall matrix.

# MODULE FUNCTIONAL DESCRIPTIONS

## 4.87.4.14 Piecewise Linear Analysis Calculations (Subroutines PSTRM and PSQDM of Module PLA3 and Subroutines PKTRM and PKQDM of Module PLA4)

The additional ECPTNL and ESTNL entries are:

$\epsilon_o^*$  - The previously computed strain value once removed.

$\epsilon^*$  - The previously computed strain value.

$E^*$  - The previously computed modulus of elasticity.

$\sigma_x^*$   
 $\sigma_y^*$   
 $\sigma_{xy}^*$

} The previously computed membrane stresses

All of the above values are initially zero with the exception of  $E^*$ , which is initially the original modulus of elasticity present on a MAT1 card.

For both PLA3 and PLA4, the element stress matrix calculations are generated in the same manner as Equation 24 of section 4.87.4.6 (Equations 79 through 82 of section 4.87.4.8 for the QDMEM), with the exception that for all DMAP loops (of the Piecewise Linear Analysis Rigid Format) after the first, the 3 by 3 material properties matrix  $[G_e]$  is replaced by a stress-dependent 3 by 3 material properties matrix  $[G_p]$  defined as follows

$$[G_p] = E_o \begin{bmatrix} 1+s_x^2 F & -v+s_x s_y F & 2\sigma_{xy}^* s_x F \\ & 1+s_y^2 F & 2\sigma_{xy}^* s_y F \\ \text{(sym)} & & 2(1+v)+4F\sigma_{xy}^{*2} \end{bmatrix}^{-1}, \quad (97)$$

where

$$\tau_o = \left[ \sigma_x^{*2} - \sigma_x^* \sigma_y^* + \sigma_y^{*2} + 3\sigma_{xy}^{*2} \right]^{1/2}, \quad (98)$$

$$F = \frac{9(E_o - E^*)}{4\tau_o^2 E^*}, \quad (99)$$

# MODULE FUNCTIONAL DESCRIPTIONS

$$s_x = \frac{2\sigma_x^* - \sigma_y^*}{3}, \quad (100)$$

$$s_y = \frac{2\sigma_y^* - \sigma_x^*}{3}, \quad (101)$$

and  $E_0$  and  $\nu$  are the linear type 1 material properties. If  $E^* = 0$ , or  $\tau_0 = 0$ , then  $[G_p] = [0]$ .

Calculate the incremental element stresses:

$$\begin{pmatrix} \Delta\sigma_x \\ \Delta\sigma_y \\ \Delta\sigma_{xy} \end{pmatrix} = \left[ \sum_{i=a,b,c} [S_i] \{\Delta u_{gi}\} \right], \quad (102)$$

where  $[S_i]$  is given in Equation 24 of section 4.87.4.6, and  $\{\Delta u_{gi}\}$  are the 3 by 1 translational displacement vectors.

Define the element stresses for output and for updating the ECPTNL and ESTNL:

$$\sigma_{x1} = \sigma_x^* + \Delta\sigma_x, \quad (103)$$

$$\sigma_{y1} = \sigma_y^* + \Delta\sigma_y, \quad (104)$$

$$\sigma_{xy1} = \sigma_{xy}^* + \Delta\sigma_{xy}. \quad (105)$$

In PLA3, using the element stresses above, the principal stresses are calculated in the same manner as in Equations 28 through 31 in section 4.87.4.6.

Estimate the next elastic coefficients as defined by the following equations:

$$\tau_1 = \left[ \sigma_{x1}^2 - \sigma_{x1}\sigma_{y1} + \sigma_{y1}^2 + 3\sigma_{xy1}^2 \right]^{1/2}, \quad (106)$$

$$\epsilon_1 = f^{-1}(\tau_1), \quad (107)$$

where  $f$  is the tabular stress-strain function. (When  $\tau_1$  is outside the range of the function, define  $E_1 = 0$ ,  $\epsilon_1 = \epsilon^*$ , and  $\epsilon^* = \epsilon_0^*$ ).



# MODULE FUNCTIONAL DESCRIPTIONS

Calculate:

$$\Delta \epsilon = \epsilon_1 - \epsilon^* , \quad (108)$$

$$\Delta \epsilon^* = \epsilon^* - \epsilon_0^* , \quad (109)$$

$$\epsilon_2 = \epsilon_1 + \gamma(\Delta \epsilon) , \quad (110)$$

where  $\gamma$  is a load ratio parameter calculated by the module driver (PLA3 or PLA4).

Calculate:

$$\tau_2 = f(\epsilon_2) , \quad (111)$$

where  $f$  is the tabular stress-strain function.

Then the estimated next modulus of elasticity,  $E_1$ , is given by:

$$E_1 = \begin{cases} \frac{\tau_2 - \tau_1}{\epsilon_2 - \epsilon_1} , & \text{for } \epsilon_2 \neq \epsilon_1 \\ 0 , & \text{for } \epsilon_2 = \epsilon_1 . \end{cases} \quad (112)$$

The new ESTNL and ECPTNL entries are:

$$\epsilon_0^* = \epsilon^* , \quad (113)$$

$$\epsilon^* = \epsilon_1 , \quad (114)$$

$$E^* = E_1 , \quad (115)$$

$$\sigma_x^* = \sigma_{x1} , \quad (116)$$

$$\sigma_y^* = \sigma_{y1} , \quad (117)$$

$$\sigma_{xy}^* = \sigma_{xy1} . \quad (118)$$

In module PLA4, the element stiffness matrices are calculated in the same manner as Equations 15 and 16 of section 4.87.4.3 (or as in section 4.87.4.9 for the QDMEM), with the

## MODULE FUNCTIONAL DESCRIPTIONS

exception that  $[G_e]$  matrix is replaced by the  $[G_p]$  matrix (Equation 97). In the calculation for  $[G_p]$ ,  $E_1$  (Equation 112) is used for  $E^*$ , and the newly calculated membrane stresses (Equations 103, 104, and 105) are used in place of  $\sigma_x^*$ ,  $\sigma_y^*$ , and  $\sigma_{xy}^*$ .

### 4.87.4.15 Thermal Analysis Calculations for the Membrane Elements TRMEM and QDMEM

If the subroutines are to be used for thermal analysis, word 56 in labeled COMMON/SYSTEM/ is +1. The geometry of the element is processed, as with a structural analysis problem, to produce the parameters  $x_2$ ,  $x_3$ ,  $y_3$ ,  $A$ , and  $t$ . For thermal analysis, the  $2 \times 2$  material conductivity matrix  $[G_e^t]$  is obtained by calling subroutine HMAT. The transformation matrices ( $2 \times 1$ ) between temperatures at the connected grid points and thermal gradients are:

$$[C_1^t] = \begin{bmatrix} -\frac{1}{x_2} \\ \frac{1}{y_3} \left( \frac{x_3}{x_2} - 1 \right) \end{bmatrix},$$

$$[C_2^t] = \begin{bmatrix} \frac{1}{x_2} \\ -\frac{x_3}{x_2 y_3} \end{bmatrix},$$

$$[C_3^t] = \begin{bmatrix} 0 \\ \frac{1}{y_3} \end{bmatrix}.$$

The scalar heat conduction terms generated in subroutine KTRMEM and placed in the KGG matrix are:

$$k_{ij}^t = A t [C_i^t]^T [G_e^t] [C_j^t],$$

where  $i$  corresponds to the pivot grid point, and  $j = 1, 2$ , and  $3$ . The term is placed in column  $SIL_i$  and row number  $SIL_j$  of the matrix.

The heat transfer "mass" matrix is generated by subroutine MTRQD in module SMA2. The thermal capacity coefficient  $C_p$  is generated by subroutine HMAT. For each triangle, the scalar terms placed in the BGG matrix are:

$$B_{ii} = \frac{A t C_p}{3} \quad i = SIL_1, SIL_2, SIL_3.$$



# STRUCTURAL ELEMENT DESCRIPTIONS

The quadrilateral is subdivided into four overlapping triangles. The mass terms for each triangle are divided by two and added to the appropriate term in the BGG matrix.

The heat transfer "stress" calculations are executed by subroutines SDHTF1, SDHTFF, and SDHTF2. SDHTF1 rearranges the EST data for all elements to a common format and calls subroutine HMAT to produce the  $2 \times 2$  conductivity matrix  $[G_e^t]$  in element coordinates. Subroutine SDHTFF calculates the Phase 1 output terms with the following calculations:

For each triangle, the  $2 \times 3$  matrix  $C_e$  is calculated where:

$$[C_e] = [C_a^t \ C_b^t \ C_c^t] = \frac{1}{2A} \begin{bmatrix} (\bar{y}_b - \bar{y}_c) & (\bar{y}_c - \bar{y}_a) & (\bar{y}_a - \bar{y}_b) \\ (\bar{x}_c - \bar{x}_b) & (\bar{x}_a - \bar{x}_c) & (\bar{x}_b - \bar{x}_a) \end{bmatrix},$$

where  $\bar{x}_i$  and  $\bar{y}_i$  are relative coordinates in the element coordinate system. For the triangle  $a = 1, b = 2, c = 3$ . For the quadrilateral, the element is broken into four triangles where  $(a, b, c) = (1, 2, 3), (2, 3, 4), (3, 4, 1),$  or  $(4, 1, 2)$ . The matrices are superimposed in the quadrilateral to produce the average  $2 \times 4$  output matrix  $[C_e]$  where

$$\begin{aligned} [C_e] = & \frac{1}{4} [C_a^t \ C_b^t \ C_c^t \ 0] \text{ triangle No. 1} \\ & + \frac{1}{4} [0 \ C_a^t \ C_b^t \ C_c^t] \text{ triangle No. 2} \\ & + \frac{1}{4} [C_c^t \ 0 \ C_a^t \ C_b^t] \text{ triangle No. 3} \\ & + \frac{1}{4} [C_b^t \ C_c^t \ 0 \ C_a^t] \text{ triangle No. 4} \end{aligned}$$

The output of Phase 1 consists of the matrix  $[C_e]$  and the material matrix  $[G_e^t]$ . The Phase 2 routine, SDHTF2, calculates the gradients,  $\begin{Bmatrix} \Delta T_x \\ \Delta T_y \end{Bmatrix}$ , and the flux vector,  $\{q\}$ , where

$$\begin{Bmatrix} \Delta T_x \\ \Delta T_y \end{Bmatrix} = [C_e] \begin{Bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{Bmatrix}$$

$$\{q\} = -[G_e^t] \begin{Bmatrix} \Delta T_x \\ \Delta T_y \end{Bmatrix}$$



# STRUCTURAL ELEMENT DESCRIPTIONS

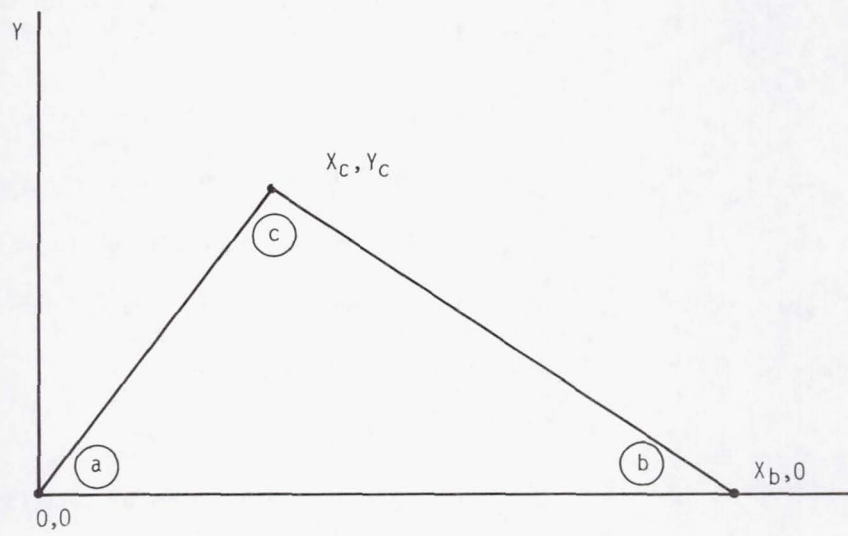


Figure 4. Triangular membrane element.

## MODULE FUNCTIONAL DESCRIPTIONS

### 4.87.5 The TRBSC, TRPLT and QDPLT Elements

#### 4.87.5.1 Input Data for the TRBSC and TRPLT Elements.

1. The ECPT/EST entries for the TRBSC and TRPLT are:

| <u>Symbol</u>                            | <u>Description</u>  |
|--|---|
| $SIL_1, SIL_2, SIL_3$                    | Scalar indices for the connected grid points  |
| $N_i, X_i, Y_i, Z_i \}$<br>$i = 1, 2, 3$ | Reference numbers for local coordinate system and locations in basic coordinates of the three connected grid points |
| $I$                                      | Bending moment of inertia per unit width  |
| $t$                                      | Effective thickness for transverse shear  |
| Mat Id. <sub>b</sub>                     | Material property identification number for bending   |
| Mat. Id. <sub>s</sub>                    | Material property identification number for shear   |
| $\theta$                                 | Material orientation angle  |
| $\mu$                                    | Nonstructural mass per area   |
| $Z_1, Z_2$                               | Fiber distances for stress calculations   |
| $t_\mu$                                  | Temperature of element for material properties  |

2. ECPT entries for the QDPLT.

The entries are the same as those for the TRPLT except that four points are used.

3. Coordinate system data

Using  $N_i, X_i, Y_i, Z_i$  and the CSTM data block the 3 by 3 global-to-basic coordinate transformation matrices  $[T_i]$  are produced for each point  $i$  via subroutines TRANSD or TRANSS.

4. Material data

Using the material property identification numbers, the orientation angle, the element temperature and the MPT and DIT data blocks, the following data are calculated:

# STRUCTURAL ELEMENT DESCRIPTIONS

| <u>Symbol</u>  | <u>Description</u>  |
|--|---|
| for Mat Id <sub>b</sub> $\begin{cases} [G_b] \\ g_e \end{cases}$ | 3x3 elastic property matrix<br>Structural damping coefficient |
| for Mat Id <sub>s</sub> $G_s$                                    | Shear coefficient   |

For the TRPLT and the QDPLT, the orientation of the material relative to each sub-triangle must be calculated from the geometry and the orientation given for the whole element. The details of this calculation are given in the next section.

## 4.87.5.2 General Calculation for the TRBSC Element

1. The coordinate system is defined by the three connected points a, b and c. {i}, {j} and {k} are the unit vectors along the x, y and z axis in basic coordinates. X<sub>i</sub>, Y<sub>i</sub> and Z<sub>i</sub> are the location coordinates of the three points, i=a, b, c. (The element coordinate system for the basic bending triangle is shown in Figure 2 of section 5.8 of the Theoretical Manual).

$$\{V_{ab}\} = \begin{Bmatrix} X_b - X_a \\ Y_b - Y_a \\ Z_b - Z_a \end{Bmatrix}, \quad (1)$$

$$\{V_{ac}\} = \begin{Bmatrix} X_c - X_a \\ Y_c - Y_a \\ Z_c - Z_a \end{Bmatrix}, \quad (2)$$

The x axis is defined by the unit vector:

$$\{i\} = \frac{\{V_{ab}\}}{|\{V_{ab}\}|}. \quad (3)$$

Calculate:

$$\{k\} = \frac{\{i\} \times \{V_{ac}\}}{|\{i\} \times \{V_{ac}\}|}. \quad (4)$$



# MODULE FUNCTIONAL DESCRIPTIONS

The y axis is defined by the unit vector,

$$\{j\} = \{k\} \times \{i\} . \quad (5)$$

2. The locations of the points in element coordinates are:

$$x_a = y_a = z_a = 0 , \quad (6)$$

$$x_b = |\{V_{ab}\}| , \quad (7)$$

$$x_c = \{V_{ac}\}^T \{i\} , \quad (8)$$

$$y_c = \{V_{ac}\}^T \{j\} = |\{i\} \times \{V_{ac}\}| . \quad (9)$$

3. The 3 x 6 transformation matrix from the six displacements in element coordinates to the three degrees of freedom used in the plate is:

$$[E]^T = \begin{bmatrix} k_1 & k_2 & k_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & i_1 & i_2 & i_3 \\ 0 & 0 & 0 & j_1 & j_2 & j_3 \end{bmatrix} . \quad (10)$$

4. The coefficients used for the plate are:

$$[D] = I[G_b] , \quad (11)$$

$$[G_2] = t_s \begin{bmatrix} G_s & 0 \\ 0 & G_s \end{bmatrix} . \quad (12)$$

# STRUCTURAL ELEMENT DESCRIPTIONS

where  $[G_b]$  is the  $3 \times 3$  material matrix for bending and  $G_s$  is the coefficient for transverse shear. The area of the triangle,  $A$ , the locations of the c.g.,  $\bar{x}$  and  $\bar{y}$ , and the radii of gyration about the origin,  $\rho_x^2$ ,  $\rho_y^2$  and  $\rho_{xy}^2$  are given by:

$$A = \frac{1}{2} x_b y_c, \quad (13)$$

$$\bar{x} = \frac{1}{3} (x_c + x_b), \quad (14)$$

$$\bar{y} = \frac{1}{3} y_c, \quad (15)$$

$$\rho_x^2 = \frac{1}{6} (x_b^2 + x_b x_c + x_c^2), \quad (16)$$

$$\rho_y^2 = \frac{1}{6} y_c^2, \quad (17)$$

$$\rho_{xy}^2 = \frac{y_c}{12} (x_b + 2x_c). \quad (18)$$

5. The stiffness matrix in generalized coordinates  $\{q\}$  is:

$$[K^X] = 4A \begin{bmatrix} D_{11} & D_{13} & D_{12} & 3\bar{x}D_{11} & \bar{x}D_{12} & 3\bar{y}D_{12} \\ & D_{33} & D_{23} & 3\bar{x}D_{13} & \bar{x}D_{23} & 3\bar{y}D_{23} \\ & & D_{22} & 3\bar{x}D_{12} & \bar{x}D_{22} & 3\bar{y}D_{22} \\ \text{SYMMETRICAL} & & & 9\rho_x^2 D_{11} & 3\rho_x^2 D_{12} & 9\rho_{xy}^2 D_{12} \\ & & & & \rho_x^2 D_{22} + 4\rho_{xy}^2 D_{23} + 4\rho_y^2 D_{33} & 3\rho_{xy}^2 D_{22} + 6\rho_y^2 D_{23} \\ & & & & & 9\rho_y^2 D_{22} \end{bmatrix} \quad (19)$$

# MODULE FUNCTIONAL DESCRIPTIONS

6. The transformation from generalized coordinates to grid point displacements (relative to point "a" of the triangle) with no transverse shear is:

$$[\bar{H}] = \begin{bmatrix} x_b^2 & 0 & 0 & x_b^3 & 0 & 0 \\ 0 & x_b & 0 & 0 & 0 & 0 \\ -2x_b & 0 & 0 & -3x_b^2 & 0 & 0 \\ x_c^2 & x_c y_c & y_c^2 & x_c^3 & x_c y_c^2 & y_c^3 \\ 0 & x_c & 2y_c & 0 & 2x_c y_c & 3y_c^2 \\ -2x_c & -y_c & 0 & -3x_c^2 & -y_c^2 & 0 \end{bmatrix} \quad (20)$$

where  $\{q\} = [\bar{H}]\{u\}$  (no transverse shear)

7. If transverse shear effects are to be calculated ( $G_s t_s \neq 0$ ), the following steps are followed:

a. Define

$$[J] = [G_2]^{-1}. \quad (21)$$

b. Calculate the transformation matrix of the shear coordinates:

$$[H_{\gamma q}] = - \begin{bmatrix} 0 & 0 & 0 & 6(J_{11}D_{11}+J_{12}D_{13}) & J_{11}(2D_{12}+4D_{33})+6J_{12}D_{23} & 6(J_{11}D_{23}+J_{12}D_{22}) \\ 0 & 0 & 0 & 6(J_{12}D_{11}+J_{22}D_{13}) & J_{12}(2D_{12}+4D_{33})+6J_{22}D_{23} & 6(J_{12}D_{23}+J_{22}D_{22}) \end{bmatrix}, \quad (22)$$



# STRUCTURAL ELEMENT DESCRIPTIONS

where

$$\begin{Bmatrix} \gamma_x \\ \gamma_y \end{Bmatrix} = [H_{\gamma q}] \{q\} . \quad (23)$$

c. The stiffness matrix of the shear terms is added to the bending stiffness matrix.

$$[K^q] = [K^X] + A t_s [H_{\gamma q}]^T [G_2] [H_{\gamma q}] . \quad (24)$$

d. The effects of shear deflection on the transformation from general to displacement coordinates is calculated:

$$[H] = [\bar{H}] - \begin{bmatrix} 0 & 0 & 0 & x_b H_{\gamma q 14} & x_b H_{\gamma q 15} & x_b H_{\gamma q 16} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_c H_{\gamma q 14} & x_c H_{\gamma q 15} & x_c H_{\gamma q 16} \\ & & & +y_c H_{\gamma q 24} & +y_c H_{\gamma q 25} & +y_c H_{\gamma q 26} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} . \quad (25)$$

If no shear exists,  $[H] = [\bar{H}]$

8. The matrix  $[H]$  is inverted:

$$[H_{qu}] = [H]^{-1} . \quad (26)$$

9. The rigid body effects are given by the matrices  $[B_b]$  and  $[B_c]$  defined as follows:

# MODULE FUNCTIONAL DESCRIPTIONS

$$[B_b] = \begin{bmatrix} 1 & 0 & -x_b \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (27)$$

$$[B_c] = \begin{bmatrix} 1 & y_c & -x_c \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (28)$$

10. The 3x3 stiffness matrix partitions in element coordinates are calculated as follows:

$$[K] = [H^{-1}]^T [K^q] [H^{-1}], \quad (29)$$

$$[K] \Rightarrow \begin{bmatrix} k_{bb} & k_{bc} \\ k_{cb} & k_{cc} \end{bmatrix}, \quad (30)$$

$$[k_{ca}] = -[k_{cb}] [B_b] - [k_{cc}] [B_c], \quad (31)$$

$$[k_{ba}] = -[k_{bb}] [B_b] - [k_{bc}] [B_c], \quad (32)$$

$$[k_{aa}] = -[B_b]^T [k_{ba}] - [B_c]^T [k_{ca}], \quad (33)$$

$$[k_{ac}] = [k_{ca}]^T, \quad (34)$$

$$[k_{ab}] = [k_{ba}]^T. \quad (35)$$

## 4.87.5.3 Stiffness Matrix Calculations for the TRBSC Element (Subroutine KTRBSC of Module SMA1).

1. If the basic triangle is used by itself as a TRBSC element, the stiffness matrices are:

$$[K_{ij}] = \begin{bmatrix} T_i^T & 0 \\ 0 & T_i^T \end{bmatrix} [E] [k_{ij}] [E]^T \begin{bmatrix} T_j & 0 \\ 0 & T_j^T \end{bmatrix}, \quad (36)$$

## STRUCTURAL ELEMENT DESCRIPTIONS

where

$$i = a, b \text{ and } c$$

$$j = a, b \text{ and } c$$

2. The structural damping matrices are calculated using  $g_e$ , the structural damping coefficient, for the referenced bending material. The 6 by 6 damping matrix partitions are:

$$[K_{ij}^4] = g_e [K_{ij}] \quad (37)$$

### 4.87.5.4 Stress Calculations for the TRBSC Element.

The stress calculations involve two phases. The first phase is used to calculate the matrix relations between element forces and grid point displacements.

1. The relation between element forces and generalized coordinates is:

$$[k_s] = \begin{bmatrix} 2D_{11} & 2D_{13} & 2D_{12} & 6\bar{x}D_{11} & 2\bar{x}D_{12}^+ & 6\bar{y}D_{12} \\ 2D_{12} & 2D_{23} & 2D_{22} & 6\bar{x}D_{12} & 2\bar{x}D_{22}^+ & 6\bar{y}D_{22} \\ 2D_{13} & 2D_{33} & 2D_{23} & 6\bar{x}D_{13} & 2\bar{x}D_{23}^+ & 6\bar{y}D_{23} \\ 0 & 0 & 0 & -6D_{11} & -2D_{12}^- & -6D_{23} \\ 0 & 0 & 0 & -6D_{13} & 4D_{33} & -6D_{22} \end{bmatrix}, \quad (38)$$

where

$$\begin{Bmatrix} M_x \\ M_y \\ M_{xy} \\ V_x \\ V_y \end{Bmatrix} = [K_s] \{q\} \quad (39)$$



# MODULE FUNCTIONAL DESCRIPTIONS

Note: When the basic triangle routine is used for stress recovery in the TRPLT or QDPLT, the values  $\bar{x}$  and  $\bar{y}$  in the above matrix are replaced by  $x_c$  and  $y_c$  or  $x_q$  and  $y_q$ .

2. The matrix  $[H]$  is calculated and inverted. The  $[B]$  matrix is calculated ( $[B]$  is a  $6 \times 3$  matrix, the  $[B_b]$  matrix (Equation 27) comprising the first three rows and the  $[B_c]$  matrix, Equation 28, comprising the last three rows). The  $[E]$  matrix and the global-to-basic transformation  $[T_a]$ ,  $[T_b]$ ,  $[T_c]$  are generated.  $[H]^{-1}$  is partitioned.

$$[H]^{-1} = [H_{1b} \mid H_{1c}]. \quad (40)$$

The element force - global displacement matrices are:

$$[S_a] = -[k_s][H]^{-1}[B][E]^T \begin{bmatrix} T_a & 0 \\ 0 & T_a \end{bmatrix}, \quad (41)$$

$$[S_b] = [k_s][H_{1b}][E]^T \begin{bmatrix} T_b & 0 \\ 0 & T_b \end{bmatrix}, \quad (42)$$

$$[S_c] = [k_s][H_{1c}][E]^T \begin{bmatrix} T_c & 0 \\ 0 & T_c \end{bmatrix}. \quad (43)$$

$$\{S_t\} = [D]\{\alpha\} \quad (43a)$$

where  $\alpha$  is the vector of thermal expansion coefficients for the bending material.

3. The second stage of stress calculations involves the use of the displacement vectors  $\{u_a\}$ ,  $\{u_b\}$  and  $\{u_c\}$ . The element forces are:

$$\begin{pmatrix} M_x \\ M_y \\ M_{xy} \\ V_x \\ V_y \end{pmatrix} = \sum_{i=a,b,c} [S_i]\{u_i\} - \{M_e\}, \quad (44)$$

# MODULE FUNCTIONAL DESCRIPTIONS

where  $\{M_e\}$  is the vector of thermal moments given on a TEMPP2 data field or if the gradient is given:

$$\{M_e\} = -T' \{S_t\} \quad .$$

With no thermal loads the stresses are:

$$\begin{Bmatrix} \sigma_{xi} \\ \sigma_{yi} \\ \sigma_{xyi} \end{Bmatrix} = -\frac{Z_i}{I} \begin{Bmatrix} M_x \\ M_y \\ M_{xy} \end{Bmatrix}, \quad i = 1, 2 \quad (45)$$

With direct thermal bending moments,  $\{M_e\}$ , given, the stresses are:

$$\begin{Bmatrix} \sigma_{xi} \\ \sigma_{yi} \\ \sigma_{xyi} \end{Bmatrix} = -\frac{Z_i}{I} \begin{Bmatrix} M_x \\ M_y \\ M_{xy} \end{Bmatrix} + \{M_e\} - \frac{1}{I} (T_i - \bar{T}) \{S_t\}, \quad i = 1, 2 \quad (45a)$$

With thermal gradient data,  $T'$ , the stresses are:

$$\begin{Bmatrix} \sigma_{xi} \\ \sigma_{yi} \\ \sigma_{xyi} \end{Bmatrix} = -\frac{Z_i}{I} \begin{Bmatrix} M_x \\ M_y \\ M_{xy} \end{Bmatrix} - \frac{1}{I} (T_i - Z_i T' - \bar{T}) \{S_t\}, \quad i = 1, 2 \quad (45b)$$

$T_i$  is the given temperature at point  $i$  and  $\bar{T}$  is the average temperature of the element. If no  $T_i$  values are given, Equation (45) is used.

## STRUCTURAL ELEMENT DESCRIPTIONS

The principal stresses and their orientation are calculated in the same manner as those for the TRMEM element, Section 4.87.4.6.

Thermal loads are generated for this element in the SSG1 module. See Section 4.87.5.12 for the equations.

### 4.87.5.6 Stiffness Matrix Calculations for the TRPLT Element (Subroutine KTRPLT of Module SMA1).

The NASTRAN bending triangle (triangular plate element, TRPLT) is fabricated from three basic bending triangles. The geometry and notation are shown in Figure 5. The general approach is to calculate the stiffness matrices for all three subtriangles or basic triangles and use the constraint equation of equal slope at the midpoints of the connected edges to calculate a reduced stiffness matrix. Since only the partitions of the stiffness matrix related to one point (the pivot grid point) are used for each calculation, the extra partitions are not used. In the NASTRAN system, the basic bending triangle calculations are in subroutine form, and the variables necessary to call it are:  $x_b$ ,  $x_c$ ,  $y_c$ , the property and material coefficients, and the transformations for orienting the anisotropic materials. The following steps are used to calculate the overall stiffness matrix for the composite triangle.

1. The element coordinate system is defined by the location of the three grid points in basic coordinates,  $\{x(1)\}$ ,  $\{x(2)\}$  and  $\{x(3)\}$ :

$$\{V_2\} = \{x(2)\} - \{x(1)\}, \quad (46)$$

$$\{V_3\} = \{x(3)\} - \{x(1)\}, \quad (47)$$

$$X_2 = |\{V_2\}|, \quad (48)$$

$$\{i\} = \frac{\{V_2\}}{X_2}, \quad (49)$$

$$Y_3 = |\{i\} \times \{V_3\}|, \quad (50)$$

$$\{k\} = \frac{\{i\} \times \{V_3\}}{Y_3}, \quad (51)$$



# MODULE FUNCTIONAL DESCRIPTIONS

$$\{j\} = \{k\} \times \{i\}, \quad (52)$$

$$[E] = \begin{bmatrix} \{k\} & 0 & 0 \\ 0 & \{i\} & \{j\} \end{bmatrix} \cdot (6 \times 3) \quad (53)$$

The locations of the points in this coordinate system are:

$$\{R(1)\} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}, \quad (54)$$

$$\{R(2)\} = \begin{Bmatrix} x_2 \\ 0 \end{Bmatrix}, \quad (55)$$

$$x_3 = \{V_3\}^T \{i\}, \quad (56)$$

$$\{R(3)\} = \begin{Bmatrix} x_3 \\ y_3 \end{Bmatrix}, \quad (57)$$

$$\{R(4)\} = \frac{1}{3} \begin{Bmatrix} x_2 + x_3 \\ y_3 \end{Bmatrix}. \quad (58)$$

2. For use in transferring points to the subtriangles, the integer matrix [M] is formed:

|       | Point a | Point b | Point c | Subtriangle $\beta$ |      |
|-------|---------|---------|---------|---------------------|------|
| [M] = | 1       | 2       | 4       | I                   | (59) |
|       | 2       | 3       | 4       | II                  |      |
|       | 3       | 1       | 4       | III                 |      |

The Roman numerals I, II and III indicate the subtriangle numbers. Points 1, 2 and 3 are the corners of the whole triangle whose centroid is denoted by 4. Points a, b, and c are the corners of the subtriangles. Point c in the subtriangles is always the center point, 4. (see Figure 5).

Note: Steps 3 through 7 are performed for each subtriangle.

# STRUCTURAL ELEMENT DESCRIPTIONS

3. The location of the three points for each subtriangle,  $\beta$ , is defined by the vectors  $\{r_i(a)\}$ ,  $\{r_i(b)\}$ ,  $\{r_i(c)\}$ . In terms of the original vectors these are:

$$\left. \begin{aligned} r_i(a) &= R_i(M(\beta,1)) \\ r_i(b) &= R_i(M(\beta,2)) \\ r_i(c) &= R_i(4) \end{aligned} \right\} \quad i = 1, 2, \quad (60)$$

$$(61)$$

$$(62)$$

where  $M(\beta, i)$  is the  $(\beta, i)$  element of the  $[M]$  matrix.

4. The variables necessary to calculate a basic bending triangle are  $x_b$ ,  $x_c$  and  $y_c$  since the local coordinate system for each subtriangle is chosen such that the "a" point lies on the origin and the "b" point lies on the x axis.

For each triangle the following are calculated:

$$\ell = \sqrt{[r_1(b) - r_1(a)]^2 + [r_2(b) - r_2(a)]^2} \quad (\text{length of base}), \quad (63)$$

$$w_1 = \frac{1}{\ell} (r_1(b) - r_1(a)), \quad (64)$$

$$w_2 = \frac{1}{\ell} (r_2(b) - r_2(a)). \quad (65)$$

5. The matrix  $[T]$  used for transforming the element coordinates to subtriangle coordinates is formed:

$$[T] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & w_1 & w_2 \\ 0 & -w_2 & w_1 \end{bmatrix}. \quad (66)$$

The material orientation angle,  $\theta_m$ , is calculated for each subtriangle. The equations are:

$$\sin(\theta_m) = w_1 \sin(\theta) - w_2 \cos(\theta), \quad (67)$$

$$\cos(\theta_m) = w_1 \cos(\theta) + w_2 \sin(\theta). \quad (68)$$

# MODULE FUNCTIONAL DESCRIPTIONS

The displacements in the subtriangle system are equal to  $[T]$  times the displacements in the original system (note  $\det [T]$  is unit length,  $x_b = R_1(2)$ ,  $x_c = R_1(4)$  and  $y_c = R_2(4)$  for subtriangle I).

6. The parameters  $x_b$ ,  $x_c$  and  $y_c$  are then computed:

$$x_b = w_1[r_1(b) - r_1(a)] + w_2[r_2(b) - r_2(a)] , \quad (69)$$

$$x_c = w_1[r_1(c) - r_1(a)] + w_2[r_2(c) - r_2(a)] , \quad (70)$$

$$y_c = -w_2[r_1(c) - r_1(a)] + w_1[r_2(c) - r_2(a)] . \quad (71)$$

7. The stiffness matrices are formed as in the basic bending triangle (Equations 30 through 35) and give:

$$[k_{ia}], [k_{ib}], [k_{ic}], [k_{ca}], [k_{cb}], [k_{cc}] \quad i = \text{pivot grid point.}$$

They relate forces and displacements in the subtriangle coordinate system and are transformed to the overall element coordinate system (i.e., the same system as subtriangle I).

Since the stiffness matrices for each pivot grid point are calculated separately, not all of these partitions are used. For each pivot grid point,  $i$ , the following partitions are used:

$$[K_{i1}], [K_{i2}], [K_{i3}] ,$$

and

$$[K_{14}], [K_{24}], [K_{34}], [K_{44}] .$$

The integer mapping matrix  $[M]$  is used to determine if and where to add the partitions. The steps used for pivot point  $i$  and triangle  $\beta$  are:



# STRUCTURAL ELEMENT DESCRIPTIONS

a)  $[T]^T [k_{ac}] [T]$  is added to  $[K_{m,4}]$ ,  $m = M(\beta,1)$

$[T]^T [k_{bc}] [T]$  is added to  $[K_{m,4}]$ ,  $m = M(\beta,2)$

$[T]^T [k_{cc}] [T]$  is added to  $[K_{4,4}]$

b) If  $M(\beta,1) = i$ :

$[T]^T [k_{aa}] [T]$  is added to  $[K_{ii}]$

$[T]^T [k_{ab}] [T]$  is added to  $[K_{i\ell}]$ ,  $\ell = M(\beta,2)$

or if  $M(\beta,2) = i$

$[T]^T [k_{bb}] [T]$  is added to  $[K_{ii}]$

$[T]^T [k_{ab}]^T [T]$  is added to  $[K_{i\ell}]$ ,  $\ell = M(\beta,1)$

c) The above is repeated for each of the three subtriangles.

8. The number 4 point in the middle is a dummy point, and since the displacements at point 4 are functions of the other displacements, it will be removed from the problem by including the calculations for the point 4 displacements in the calculations for the corner displacements, as shown in steps 8a, 8b, 8c and 8d.

a. Calculate the following geometric constants:

$$\ell_1 = \left( x_c^2 + y_c^2 \right)^{1/2}, \quad (72)$$

$$\ell_2 = \left[ (x_b - x_c)^2 + y_c^2 \right]^{1/2}, \quad (73)$$

$$s_1 = \frac{x_c}{\ell_1}, \quad (74)$$

$$s_2 = \frac{x_b - x_c}{\ell_2}, \quad (75)$$

$$c_1 = \frac{y_c}{\ell_1}, \quad (76)$$

# MODULE FUNCTIONAL DESCRIPTIONS

$$c_2 = \frac{y_c}{x_2} \quad (77)$$

b. The formulas for the locations of the midpoints are:

$$x_1 = \frac{1}{2} x_c \quad (78)$$

$$y_1 = \frac{1}{2} y_c \quad (79)$$

$$x_2 = \frac{x_b + x_c}{2} \quad (80)$$

$$y_2 = \frac{1}{2} y_c \quad (81)$$

c. The  $[H_\psi]$  matrix is calculated as follows:

$$[H_\psi] = \begin{bmatrix} -2x_1 c_1 (x_1 s_1 - y_1 c_1) & 2y_1 s_1 & -3x_1^2 c_1 & y_1 (2x_1 s_1 - y_1 c_1) & 3y_1^2 s_1 \\ 2x_2 c_2 (x_2 s_2 + y_2 c_2) & 2y_2 s_2 & 3x_2^2 c_2 & y_2 (2x_2 s_2 + y_2 c_2) & 3y_2^2 s_2 \end{bmatrix} \quad (82)$$

The slopes in terms of the deflections of points a, b and c are defined by the matrices:

$$[\bar{H}_{\psi b} \mid \bar{H}_{\psi c}] = [H_\psi][H]^{-1} \quad (83)$$

$$[\bar{H}_{\psi a}] = -[H_\psi][H]^{-1} [B] + \begin{bmatrix} 0 & s_1 & c_1 \\ 0 & s_2 & c_2 \end{bmatrix} \quad (84)$$

where  $[H]^{-1}$  and  $[B]$  are calculated while generating the stiffness matrices.  $[H]^{-1}$  is the inverse of the 6x6 matrix in Equation 25, and  $[B] = \begin{bmatrix} B_b \\ B_c \end{bmatrix}$  is a 6x3 matrix where  $[B_b]$  and  $[B_c]$  are calculated in Equations 27 and 28 respectively.

These are transformed to element coordinates by:

$$[H_{\psi\alpha}] = [\bar{H}_{\psi\alpha}] [T] \quad (85)$$

d. Each row of the matrix corresponds to the slope angle of the mid-point of the sides.



## STRUCTURAL ELEMENT DESCRIPTIONS

The first row defines the slope of the normal to the line connecting point "a" to the center point. The second row defines the slope of the normal to the line connecting point "b" to the center point. Using the  $[H_{\psi\alpha}]$  matrices, four 3 by 3 matrices,  $[G_M]$ , are formed as follows:

The  $[M]$  matrix is now used:

$$[M] = \begin{matrix} & \overbrace{\begin{matrix} a & b & c \end{matrix}}^{\alpha} \\ \begin{bmatrix} 1 & 2 & 4 \\ 2 & 3 & 4 \\ 3 & 1 & 4 \end{bmatrix} & \begin{matrix} I \\ II \\ III \end{matrix} & = \beta \end{matrix}, \quad (86)$$

For the  $[H_{\psi\alpha}]^\beta$  matrix, ( $\alpha = a, b$  and  $c$ ;  $\beta = I, II$  or  $III$ ), the number  $M(\beta, \alpha)$  which identifies the matrix  $[G_M]$  is found. The three terms in the first row of  $[H_{\psi\alpha}]^\beta$  are added to the  $M(\beta, 1)$  row of the  $[G_M]$  matrix. The three numbers in the second row of  $[H_{\psi\alpha}]^\beta$  are added to the  $M(\beta, 2)$  row of the  $[G_M]$  matrix. This procedure is repeated for the three  $[H_{\psi\alpha}]^\beta$  matrices for each triangle  $\beta$ .

The stiffness matrix partitions of the whole plate are computed from:

$$\begin{aligned} [K_{ij}^e] &= [K_{ij}] - ([G_4]^{-1}[G_i])^T [K_{j4}]^T - [K_{i4}] [G_4]^{-1} [G_j] \\ &+ ([G_4]^{-1}[G_i])^T [K_{44}] [G_4]^{-1} [G_j] \quad \text{for } i = 1, 2, 3, \\ &\quad j = 1, 2, 3. \end{aligned} \quad (87)$$

where  $[K_{ij}]$  are computed as in step 7.

9. Using the locations of the three grid points in basic coordinates, the 3x3 transformation matrices  $[T_j]$ ,  $j = 1, 2, 3$ , are calculated. The 6x6 matrices  $[C_j]$  are formed as:

$$[C_j] = \begin{bmatrix} [T_j] & \vdots & 0 \\ 0 & \vdots & [T_j] \end{bmatrix} \quad j = 1, 2, 3. \quad (88)$$

10. The stiffness matrix partitions in global coordinates for pivot point  $i$  are computed from:



# MODULE FUNCTIONAL DESCRIPTIONS

$$[K_{ij}^g] = [C_i]^T [E] [K_{ij}^e] [E]^T [C_j] \quad (6 \times 6), \quad (89)$$

where:

$[K_{ij}^e]$  was calculated in step 8

$[E]$  was calculated in step 1

$[C_i], [C_j]$  were calculated in step 9

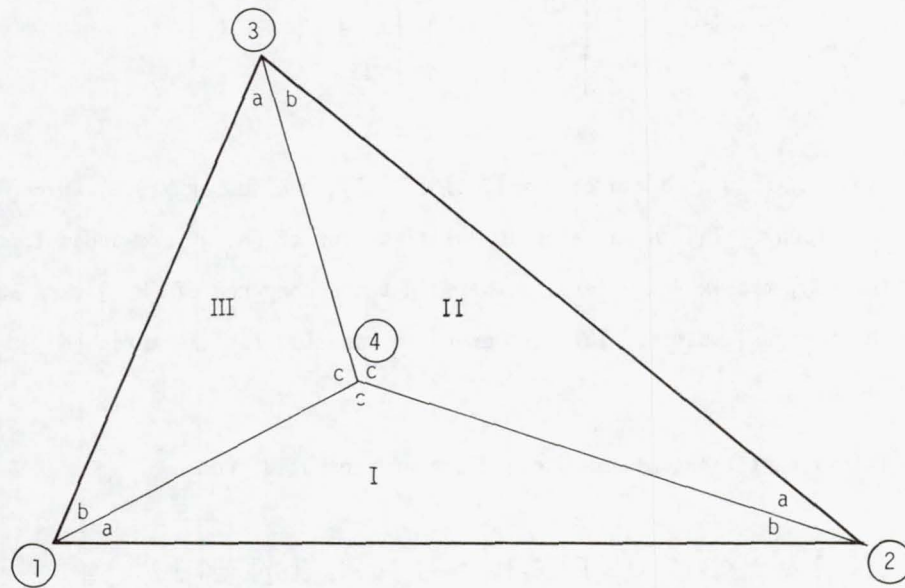


Figure 5. The triangular plate element, TRPLT.

- 1, 2, 3 = GIVEN GRID POINTS
- 4 = CENTROID OF TRIANGLE
- I, II, III = BASIC SUBTRIANGLES
- a, b, c = ORDERED VERTICES OF SUBTRIANGLES

## STRUCTURAL ELEMENT DESCRIPTIONS

### 4.87.5.6 Structural Damping Matrices for the TRPLT Element.

The structural damping matrices are:

$$[K_{ij}^4] = g_e [K_{ij}^g], \quad (90)$$

where  $g_e$  is the structural damping coefficient for the bending material referenced.

### 4.87.5.7 Stress and Element Force Calculations for the TRPLT Element (Subroutines STRPL1 and SBSPL2 of Module SDR2).

For stress recovery, the relationship between the center point and the corner points is used to describe the stress functions for each subtriangle. The stresses in each subtriangle at the center point are averaged to provide the final element stress and forces.

1. STRPL1 is used to calculate the phase I stress-displacement relations.

The following data are calculated using the same equations as those for the stiffness matrix generation routines.

$[C_i]$  -  $i = 1, 2, 3$  - Global-to-basic transformations

$[E]$  - element to basic coordinate transformation

Values computed for  
each subtriangle

$\left\{ \begin{array}{l} x_b, x_c, y_c - \text{subtriangle point locations} \\ \sin\theta, \cos\theta - \text{material orientation} \\ w_1, w_2 - \text{element-to-subtriangle coordinate transformation} \\ [\bar{H}_{\psi a}], [\bar{H}_{\psi b}], [\bar{H}_{\psi c}] - \text{normal slope angle relationship matrices} \end{array} \right.$

For each subtriangle,  $\beta = I, II, III$ , the following matrices are formed:

# MODULE FUNCTIONAL DESCRIPTIONS

$$[T^\beta] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & w_1 & w_2 \\ 0 & -w_2 & w_1 \end{bmatrix}, \quad (91)$$

$$[V^\beta] = \begin{bmatrix} w_1^2 & w_2^2 & -2w_1w_2 & 0 & 0 \\ w_2^2 & w_1^2 & 2w_1w_2 & 0 & 0 \\ w_1w_2 & -w_1w_2 & w_1^2 - w_2^2 & 0 & 0 \\ 0 & 0 & 0 & w_1 & -w_2 \\ 0 & 0 & 0 & w_2 & w_1 \end{bmatrix} \quad (92)$$

For each subtriangle  $\beta$ , three 5x3 transformations are calculated,  $[S_\alpha^\beta]$ ,  $\alpha = a, b, c$ .

These are transformed and added to the corresponding matrices for each point with the equation:

$$[S_M^*] = \frac{1}{3} \sum_{\beta} [V^\beta][S_\alpha^\beta][T^\beta], \quad (93)$$

where the  $\alpha$ , which denotes points on the subtriangle, corresponds to the grid point M on the overall triangle.

$$[H_{\psi\alpha}] = [\bar{H}_{\psi\alpha}][T] \quad \alpha = a, b, c. \quad (94)$$

The  $[H_{\psi\alpha}]$  matrices are added to the corresponding  $[G_M]$  matrix with the appropriate row interchanges. When the data for all three subtriangles have been generated, the following operations are performed:

$$[S_M^e] = [S_M^*] - [S_4^*][G_4]^{-1}[G_i] \quad (95)$$

for  $M = 1, 2, 3$ ;

$$[S_M] = [S_M^e][E]^T[C_i] \quad (96)$$

for  $M = 1, 2, 3$ .



# STRUCTURAL ELEMENT DESCRIPTIONS

## 2. Phase 2

a) The vector of forces is computed first.

$$\begin{Bmatrix} M_x \\ M_y \\ M_{xy} \\ V_x \\ V_z \end{Bmatrix} = \sum_{i=1}^3 [S_i] \{u_i\} - \{M_t\} \quad , \quad (97)$$

where  $\{M_e\}$  is the thermal moment vector. If a thermal gradient is given:

$$\{M_e\} = -\{S_t\}T' \quad .$$

b) With no given temperatures at the stress points, the stresses are then calculated from the equations

$$\sigma_{xi} = \frac{M_x Z_i}{I} \quad (98)$$

$$\sigma_{yi} = \frac{M_y Z_i}{I} \quad i = 1, 2 \quad . \quad (99)$$

$$\sigma_{xyi} = \frac{M_{xy} Z_i}{I} \quad (100)$$

If temperature values  $T_i$  at the stress points are given the following equations are used.

$$\begin{Bmatrix} \sigma_{xi} \\ \sigma_{yi} \\ \sigma_{xyi} \end{Bmatrix} = -\frac{Z_i}{I} \begin{Bmatrix} M_x \\ M_y \\ M_{xy} \end{Bmatrix} + \{M_e\} - \frac{1}{I} (T_i - \bar{T}) \{S_t\} \quad , \quad i = 1 \text{ or } 2$$

or

## STRUCTURAL ELEMENT DESCRIPTIONS

$$\begin{pmatrix} \sigma_{xi} \\ \sigma_{yi} \\ \sigma_{xyi} \end{pmatrix} = -\frac{Z_i}{I} \begin{pmatrix} M_x \\ M_y \\ M_{xy} \end{pmatrix} - \frac{1}{I} (T_i - Z_i T' - \bar{T}) \{S_t\}, \quad i = 1 \text{ or } 2$$

where  $T'$  is the gradient or  $\{M_e\}$  is the thermal moment vector.  $\bar{T}$  is the average temperature for the element. The principal stresses and angles are calculated using the same formula as those for the membrane element (see Section 4.87.4.6, Equations 28, 29 and 30).

Thermal loads are generated for this element in the SSG1 module. See Section 4.87.4.8.5.11 for the algorithm.

### 4.87.5.8 Stiffness Matrix Calculations for the QDPLT Element (Subroutine KQDPLT of Module SMA1)

The quadrilateral plate element uses two sets of overlapping triangles as shown in Figure 6. The logic is the same as that for the quadrilateral membrane except that the order of the points of the triangles is chosen to place the triangle coordinate systems along the diagonals.

1. The following equations are used to calculate the three unit vectors,  $\{i\}$ ,  $\{j\}$  and  $\{k\}$ , which define the element coordinate system.

# MODULE FUNCTIONAL DESCRIPTIONS

$$\{V_i\} = \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix}, \quad i = 1, 2, 3, 4. \quad (101)$$

The diagonals are:

$$\{d_1\} = \{V_3\} - \{V_1\}, \quad (102)$$

$$\{d_2\} = \{V_4\} - \{V_2\}. \quad (103)$$

The area is calculated from:

$$A = \frac{1}{2} |(\{d_1\} \times \{d_2\})|. \quad (104)$$

The normal to the plane is calculated from:

$$\{k\} = \frac{\{d_1\} \times \{d_2\}}{|\{d_1\} \times \{d_2\}|}, \quad (105)$$

$$\{a_1\} = \{V_2\} - \{V_1\}, \quad (106)$$

$$h = \frac{1}{2} \{a_1\}^T \{k\}, \quad (107)$$

The vectors lying in the new plane are computed from:

$$\{i\} = \frac{\{a_1\} - 2h\{k\}}{|\{a_1\} - h\{k\}|}, \quad (108)$$

$$\{j\} = \{k\} \times \{i\}. \quad (109)$$

The nonzero positions of the points in the plane are computed from:



# STRUCTURAL ELEMENT DESCRIPTIONS

$$x_2 = \{a_i\}^T \{i\} , \quad (110)$$

$$x_3 = \{d_i\}^T \{i\} , \quad (111)$$

$$y_3 = \{d_i\}^T \{j\} , \quad (112)$$

$$x_4 = x_2 + (\{d_2\}^T \{i\}) , \quad (113)$$

$$y_4 = \{d_2\}^T \{j\} , \quad (114)$$

$$\{R(i)\} = \begin{Bmatrix} x_i \\ y_i \end{Bmatrix} . \quad (115)$$

## 2. Element interior angle tests.

The interior angles of the quadrilateral must be less than  $180^\circ$ . The following checks accomplish this task.

| <u>Test</u> |   | <u>Point with angle greater than <math>180^\circ</math></u> |
|-------------|---|---|
| If          | $y_4 < 0$                                 | 1   |
| If          | $y_3 < 0$                                 | 2   |
| If          | $x_4 > x_2 - (x_2 - x_3) \frac{y_4}{y_3}$ | 3   |
| If          | $x_3 < \frac{y_3}{y_4} x_4$               | 4   |

3. The relative data for each subtriangle must be calculated and passed to the matrix calculation subroutine. The integer mapping matrix  $[M]$  denotes which points, 1, 2, 3, and 4 of the quadrilateral, are used in the subtriangle. The row position indicates the subtriangle to which the point belongs, and the column position indicates the corresponding point in that subtriangle.

# MODULE FUNCTIONAL DESCRIPTIONS

|       | <u>Point a</u> | <u>Point b</u> | <u>Point c</u> | <u>Triangle No.</u> |
|-------|----------------|----------------|----------------|---------------------|
| [M] = | 2              | 4              | 1              | I                   |
|       | 3              | 1              | 2              | II                  |
|       | 4              | 2              | 3              | III                 |
|       | 1              | 3              | 4              | IV                  |

4. For each triangle the stiffness matrix is calculated in its own coordinate system. This system has its origin at point a, point b lies on the x axis, and point c has a positive y component.  $\{R(i)\}$  values are transferred to  $\{r(\alpha)\}$  values ( $\alpha = a, b, c$ ), and the following calculations are performed:

$$\{v(b)\} = \{r(b)\} - \{r(a)\}, \quad (116)$$

$$\{v(c)\} = \{r(c)\} - \{r(a)\}, \quad (117)$$

$$\begin{Bmatrix} w_1 \\ w_2 \end{Bmatrix} = \frac{\{v(b)\}}{|\{v(b)\}|}, \quad (118)$$

$$x_b = |\{v(b)\}|, \quad (119)$$

$$\begin{Bmatrix} x_c \\ y_c \end{Bmatrix} = \begin{bmatrix} w_1 & w_2 \\ -w_2 & w_1 \end{bmatrix} \{v(c)\}. \quad (120)$$

$x_b$ ,  $x_c$  and  $y_c$  are the point locations needed by the subroutine.

$$[T] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & w_1 & w_2 \\ 0 & -w_2 & w_1 \end{bmatrix}. \quad (121)$$

## STRUCTURAL ELEMENT DESCRIPTIONS

$w_1$  is the x component of the new x axis and  $w_2$  is its y component,  $[T]$  transforms the z displacement and the two angles from the quadrilateral system to the triangle system.

In order to calculate the material matrices in the basic triangle routine, the material orientation angle,  $\theta_m$ , is :

$$\sin \theta_m = w_1 \sin \theta - w_2 \cos \theta , \quad (122)$$

$$\cos \theta_m = w_1 \cos \theta + w_2 \sin \theta . \quad (123)$$

$w_1$  and  $w_2$  are the cosine and sine of the angle made between the base of the triangle and the material orientation axis.

5. The output of the basic bending triangle routine are the 3x3 matrices:

$$[k_{aa}], [k_{ab}], [k_{bb}], [k_{ac}], [k_{bc}], [k_{cc}] .$$

To transform these to the quadrilateral system the following equation is used:

$$[K_{ij}^e] = \frac{1}{2} [T]^T [k_{ij}] [T] . \quad (124)$$

These matrices are added to the current positions in the quadrilateral matrix partitions using the  $[M]$  matrix in step 3.

6. For each pivot point i the following 3x3 partitions are formed:

$$[k_{ij}^e], \text{ for } i = 1, 2, 3, 4 .$$

7. Using the geometry data, the 3x3 global-to-basic transformations  $[T_j]$  are formed for  $j = 1, 2, 3, 4$ . These are expanded to the 6x6 matrices  $[C_j]$ :



# MODULE FUNCTIONAL DESCRIPTIONS

$$[C_j] = \begin{bmatrix} T_j & | & 0 \\ \hline 0 & | & T_j \end{bmatrix}. \quad (125)$$

8. The stiffness matrix partitions in global coordinates are found from:

$$[K_{ij}^g] = [C_i]^T [E] [k_{ij}^e] [E]^T [C_j], \quad (126)$$

where  $[E]$  is defined in Equation 53.

## 4.87.5.9 Stress and Element Force Calculations for QDPLT Element (Subroutines SQDPL1 and SBSPL2 of Module SDR2)

1. SQDPL1 calculates the phase 1 stress-displacement relations. A coordinate system matrix  $[E]$ , the subtriangle base vectors  $\{w_1, w_2\}^T$  and the global-to-basic transformation matrices are calculated with the same equations as those used in the plate element stiffness matrix calculations. For each subtriangle,  $\beta$ , the following matrices are formed:

$$[T^\beta] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & w_1 & w_2 \\ 0 & -w_2 & w_1 \end{bmatrix}, \quad (127)$$

$$[V^\beta] = \begin{bmatrix} w_1^2 & w_2^2 & -2w_1w_2 & 0 & 0 \\ w_2^2 & w_1^2 & 2w_1w_2 & 0 & 0 \\ w_1w_2 & -w_1w_2 & (w_1^2 - w_2^2) & 0 & 0 \\ 0 & 0 & 0 & w_1 & -w_2 \\ 0 & 0 & 0 & w_2 & w_1 \end{bmatrix}. \quad (128)$$

# STRUCTURAL ELEMENT DESCRIPTIONS

For each subtriangle the stresses must be calculated at the intersection point of the diagonals. In the quadrilateral system:

$$x_5 = \frac{x_2 x_3 y_4}{x_3 y_4 + (x_2 - x_4) y_3}, \quad (129)$$

$$y_5 = \frac{y_3}{x_3} x_5. \quad (130)$$

For each subtriangle:

$$\{V_5\} = \{r(a)\} - \begin{Bmatrix} x_5 \\ y_5 \end{Bmatrix}; \quad x_q = |\{V_5\}|; \quad y_q = 0; \quad (131)$$

where  $x_q$  and  $y_q$  denote the location of the intersection of the diagonals of the quadrilateral in the subtriangle coordinate system. The stresses are calculated at this point.

$[T^\beta]$  transforms the translation and two rotations in the element system to the subtriangle system.  $[V^\beta]$  transforms the three moments and two shears of the subtriangle to the element system. For each subtriangle  $\beta$ , three  $5 \times 3$  transformations are calculated,  $[S_\alpha^\beta]$ ,  $\alpha = a, b, c$ . These are transformed and added to the corresponding matrices for each point with the equation:

$$[S_M^e] = \frac{1}{4} \sum_{\beta} [V^\beta] [S_\alpha^\beta] [T^\beta], \quad (132)$$

where the  $\alpha$ , which denotes points on the subtriangle, corresponds to the grid point M on the quadrilateral.

2. Using the basic-to-element and global-to-basic transformations, the stress matrices  $[S_M]$  in global coordinates are formed from:

$$[S_M] = [S_M^e] [E]^T \begin{bmatrix} [T_i] & \begin{Bmatrix} 0 \\ [T_i] \end{Bmatrix} \\ 0 & \begin{Bmatrix} [T_i] \end{Bmatrix} \end{bmatrix}. \quad (133)$$

## MODULE FUNCTIONAL DESCRIPTIONS

The thermal load vector is:

$$\{S_t\} = [D]\{\alpha\}$$

3. The additional data for phase 2 calculations are

$I, Z_1, Z_2$  from the ECPT data.

4. Phase 2 (Subroutine SBSPL2)

The 5 by 6  $[S_M]$  matrices are used in the same manner as the TRPLT elements (see Equations 97 through 100).

- 4.87.5.10 Lumped Mass Matrix Generation for the TRBSC, TRPLT, and QDPLT Elements (Subroutine MASSTQ of Module SMA2).

The lumped mass matrices are calculated in the same manner as the triangular or quadrilateral membrane except that the material density is not used (see Equations 86 through 96 in section 4.87.4.10). The only contribution to the mass matrix from these elements is the nonstructural mass,  $\mu$ .



# MODULE FUNCTIONAL DESCRIPTIONS

## 4.87.5.11 Coupled Mass Matrix Calculations for the TRBSC Element (Subroutine MTRBSC of Module SMA2).

The mass properties of the two-dimensional elements are defined by their geometry, the mass density given by the material, the thickness of the element and the nonstructural mass. The normal execution of NASTRAN will result in the calculation of the total mass of the element and distribute it as lumped masses at the connected grid points (subroutine MASSTQ of module SMA2). If the parameter C0UPBAR is set by the user, the elements with bending properties will have their mass distributed according to their elastic properties. This results in element mass matrices with directional properties and coupling terms between points in an element. Since the thickness of the TRBSC element is zero, a coupled mass matrix for this element does not exist. The MTRBSC subroutine is used exclusively by subroutines MTRPLT and MQDPLT.

1. The mass matrix  $[\tilde{M}]$  in generalized coordinates is calculated in the following steps.

a. Integral values  $I_{ij}$  used in the mass matrices are calculated from the formulas:

$$A_{0j} = \left[ \frac{x_b}{j+1} - \frac{x_b - x_c}{j+2} \right] y_c^{j+1}, \quad j = 0, 1, \dots, 6 \quad (134)$$

$$A_{ij} = \frac{(x_c)^{i+1} (y_c)^{j+1}}{(i+1)(i+j+2)} + \frac{ix_b}{(i+j+2)} A_{i-1,j}, \quad \begin{matrix} i = 1, 2, \dots, 6 \\ j = 0, 1, \dots, 6 \end{matrix} \quad (135)$$

$$B_{ij} = - \frac{(x_c)^{i+1}}{(y_c)^{i+1} (i+1)(i+j+2)} (y_c)^{i+j+2}, \quad \begin{matrix} i = 0, 1, \dots, 6 \\ j = 0, 1, \dots, 6 \end{matrix} \quad (136)$$

$$I_{ij} = m[A_{ij} + B_{ij}], \quad \begin{matrix} i = 0, 1, \dots, 6 \\ j = 0, 1, \dots, 6 \end{matrix} \quad (137)$$

where  $m$  is the nonstructural mass, and where  $x_b$ ,  $x_c$  and  $y_c$  are computed in Equations 7, 8, and 9 respectively.

# MODULE FUNCTIONAL DESCRIPTIONS

b. Partition  $[\tilde{M}]$  into submatrices.

$$[\tilde{M}] \Rightarrow \begin{bmatrix} \bar{M}_{aa} & M_{ar} \\ M_{ar}^T & M_{rr} \end{bmatrix}, \quad (138)$$

where  $[\bar{M}_{aa}]$  is a 3 by 3 matrix,  $[M_{ar}]$  is a 3 by 6 matrix and  $[M_{rr}]$  is a 6 by 6 matrix.

c. The mass matrix  $[\bar{M}_{aa}]$  is given by:

$$[\bar{M}_{aa}] = \begin{bmatrix} I_{00} & I_{01} & -I_{10} \\ I_{01} & I_{02} & -I_{11} \\ -I_{10} & -I_{11} & I_{20} \end{bmatrix}. \quad (139)$$

d. The other matrices,  $[M_{ar}]$  and  $[M_{rr}]$ , are less simple. The algebraic expressions for the elements of these matrices are given in Tables 1a and 1b below.

Table 1a. Elements of the 3 by 6 Matrix  $[M_{ar}]$ .

Notes:  $H_{\gamma q_{ij}}$  is contracted to  $H_{ij}$ , where  $[H_{\gamma q}]$  is computed in Equation 22.

$M_{ar_{ij}}$  is contracted to  $M_{ik}$ , where  $M_{ik}$  represents an element of  $[\tilde{M}]$  given in Equation 138.

$$M_{14} = I_{20}$$

$$M_{15} = I_{11}$$

$$M_{16} = I_{02}$$

$$M_{17} = I_{30} + H_{14}I_{10} + H_{24}I_{01}$$

# MODULE FUNCTIONAL DESCRIPTIONS

Table 1a (con'd). Elements of the 3 by 6 Matrix  $[M_{ar}]$ .

$$M_{18} = I_{12} + H_{15}I_{10} + H_{25}I_{01}$$

$$M_{19} = I_{03} + H_{16}I_{10} + H_{26}I_{01}$$

$$M_{24} = I_{21}$$

$$M_{25} = I_{12}$$

$$M_{26} = I_{03}$$

$$M_{27} = I_{31} + H_{14}I_{11} + H_{24}I_{02}$$

$$M_{28} = I_{13} + H_{15}I_{11} + H_{24}I_{02}$$

$$M_{29} = I_{04} + H_{16}I_{11} + H_{26}I_{02}$$

$$M_{34} = -I_{30}$$

$$M_{35} = -I_{21}$$

$$M_{36} = -I_{12}$$

$$M_{37} = -I_{40} - H_{14}I_{20} - H_{24}I_{11}$$

$$M_{38} = -I_{22} - H_{15}I_{20} - H_{25}I_{11}$$

$$M_{39} = -I_{13} - H_{16}I_{20} - H_{26}I_{11}$$



# MODULE FUNCTIONAL DESCRIPTIONS

Table 1b. Elements of the 6 by 6 Matrix  $[M_{rr}]$ .

Notes:  $H_{Yq_{ij}}$  is contracted to  $H_{ij}$ ;  $M_{rr_{ij}}$  is contracted to  $M_{ik}$ , where  $M_{ik}$  represents an element of  $[\tilde{M}]$  given in Equation 138;  $M_{ij} = M_{ji}$ .

$$M_{44} = I_{40}$$

$$M_{45} = I_{31}$$

$$M_{46} = I_{22}$$

$$M_{47} = I_{50} + H_{14}I_{30} + H_{24}I_{21}$$

$$M_{48} = I_{32} + H_{15}I_{30} + H_{25}I_{21}$$

$$M_{49} = I_{23} + H_{16}I_{30} + H_{26}I_{21}$$

$$M_{55} = I_{22}$$

$$M_{56} = I_{13}$$

$$M_{57} = I_{41} + H_{14}I_{21} + H_{24}I_{12}$$

$$M_{58} = I_{23} + H_{15}I_{21} + H_{25}I_{12}$$

$$M_{59} = I_{14} + H_{16}I_{21} + H_{26}I_{12}$$

$$M_{66} = I_{04}$$

$$M_{67} = I_{32} + H_{14}I_{12} + H_{24}I_{03}$$

$$M_{68} = I_{14} + H_{15}I_{12} + H_{25}I_{03}$$

$$M_{69} = I_{05} + H_{16}I_{12} + H_{26}I_{03}$$

$$M_{77} = I_{60} + 2H_{14}I_{40} + 2H_{24}I_{31} + (H_{14})^2I_{20} + 2H_{14}H_{24}I_{11} + (H_{24})^2I_{02}$$

# MODULE FUNCTIONAL DESCRIPTIONS

Table 1b (con'd). Elements of the 6 by 6 Matrix  $[M_{rr}]$ .

$$M_{78} = I_{42} + H_{15}I_{40} + H_{25}I_{31} + H_{14}I_{22} + H_{14}H_{15}I_{20} + H_{14}H_{25}I_{11} + H_{24}I_{13} \\ + H_{24}H_{15}I_{11} + H_{24}H_{25}I_{02}$$

$$M_{79} = I_{33} + H_{16}I_{40} + H_{26}I_{31} + H_{14}I_{13} + H_{14}H_{16}I_{20} + H_{14}H_{26}I_{11} + H_{24}I_{04} \\ + H_{24}H_{16}I_{11} + H_{24}H_{26}I_{02}$$

$$M_{88} = I_{24} + 2H_{15}I_{22} + 2H_{25}I_{13} + (H_{15})^2I_{20} + 2H_{15}H_{25}I_{11} + (H_{25})^2I_{02}$$

$$M_{89} = I_{15} + H_{16}I_{22} + H_{26}I_{13} + H_{15}I_{13} + H_{15}H_{16}I_{20} + H_{15}H_{26}I_{11} + H_{25}I_{04} \\ + H_{25}H_{16}I_{11} + H_{25}H_{26}I_{02}$$

$$M_{99} = I_{06} + 2H_{16}I_{13} + 2H_{26}I_{04} + (H_{16})^2I_{20} + 2H_{16}H_{26}I_{11} + (H_{26})^2I_{02}$$

# MODULE FUNCTIONAL DESCRIPTIONS

2. The mass matrix  $[M_{\text{mass}}]$  in element coordinates is calculated from the following equation:

$$[M_{\text{mass}}] = \begin{bmatrix} I & 0 \\ -H^{-1}B & H^{-1} \end{bmatrix}^T \begin{bmatrix} \bar{M}_{aa} & M_{ar} \\ M_{ar}^T & M_{rr} \end{bmatrix} \begin{bmatrix} I & 0 \\ -H^{-1}B & H^{-1} \end{bmatrix}, \quad (140)$$

where  $[I]$  is a 3 x 3 identity matrix,  $[H]^{-1}$  is calculated as in Equation 25,  $[B] = \begin{bmatrix} B_b \\ B_c \end{bmatrix}$  is calculated as in Equations 27 and 28.

The calculations are broken down into the following steps where:

$$[M_{\text{mass}}] = \begin{bmatrix} M_{aa} & M_{ab} & M_{ac} \\ M_{ba} & M_{bb} & M_{bc} \\ M_{ca} & M_{cb} & M_{cc} \end{bmatrix}, \quad (141)$$

and  $[M_{ij}]$ ,  $i = a, b, c$  and  $j = a, b, c$  are 3 by 3 matrices.

a) Compute:

$$[M] = [H^{-1}]^T [M_{rr}] [H^{-1}]. \quad (142)$$

b) Partition:

$$[M] \Rightarrow \begin{bmatrix} M_{bb} & M_{bc} \\ M_{cb} & M_{cc} \end{bmatrix}. \quad (143)$$

c) Compute:

$$[M_{ai}] = [M_{ar}] [H^{-1}]. \quad (144)$$

d) Partition:

$$[M_{ai}] \Rightarrow [\bar{M}_{ab} \mid \bar{M}_{ac}]. \quad (145)$$



# MODULE FUNCTIONAL DESCRIPTIONS

e) Calculate:

$$[M_{ab}] = [\bar{M}_{ab}] - [B_b]^T [M_{bb}] - [B_c]^T [M_{cb}] \quad , \quad (146)$$

$$[M_{ac}] = [\bar{M}_{ac}] - [B_b]^T [M_{bc}] - [B_c]^T [M_{cc}] \quad , \quad (147)$$

$$[M_{aa}] = [\bar{M}_{aa}] - [B_b]^T [M_{ab}]^T - [B_c]^T [M_{ac}]^T - [\bar{M}_{ab}] [B_b] - [\bar{M}_{ac}] [B_c] \quad , \quad (148)$$

$$[M_{ba}] = [M_{ab}]^T \quad , \quad (149)$$

$$[M_{ca}] = [M_{ac}]^T \quad . \quad (150)$$

## 4.87.5.12 Mass Matrix Calculations for the TRPLT Element (Subroutine MTRPLT of Module SMA2)

1. The general calculations for the mass matrix of the triangular plate element, TRPLT, are the same as those for the stiffness matrix calculations (See Equations 46 through 71).
2. For each subtriangle the output from the basic bending triangle subroutine are the nine 3 x 3 matrices given in Equation 141:

|          |          |          |
|----------|----------|----------|
| $M_{aa}$ | $M_{ab}$ | $M_{ac}$ |
| $M_{ba}$ | $M_{bb}$ | $M_{bc}$ |
| $M_{ca}$ | $M_{cb}$ | $M_{cc}$ |

They relate forces and accelerations in the subtriangle coordinate system and must be transformed to the overall element coordinate system (i.e., the same system as subtriangle I).

The matrix partitions in the subtriangles are added to the correct matrix partition for the whole triangle. For example, for subtriangle number II,  $[M_{aa}]$  is transformed and added to  $[\bar{M}_{22}]$ ,  $[M_{ab}]$  is transformed and added to  $[\bar{M}_{23}]$ ,  $[M_{ac}]$  is transformed and added to  $[\bar{M}_{24}]$ ,  $[M_{ba}]$  is transformed and added to  $[\bar{M}_{32}]$ , etc.

# MODULE FUNCTIONAL DESCRIPTIONS

Since the mass matrices for each pivot grid point are calculated separately, not all of these partitions are used. For each pivot grid point,  $i$ , the matrices which are used will be:

$$[\bar{M}_{i1}], [\bar{M}_{i2}], [\bar{M}_{i3}] \quad , \quad i = \text{point } 1, 2 \text{ or } 3 \text{ of composite triangle}$$

and

$$[\bar{M}_{14}], [\bar{M}_{24}], [\bar{M}_{34}], [\bar{M}_{44}] \quad .$$

3. The number 4 point in the middle is a dummy point and is removed from the problem in the same manner as in the computation of the stiffness matrix (see Equations 72 through 86).

The mass matrix partitions of the whole plate are:

$$\begin{aligned} [M_{ij}^e] &= [\bar{M}_{ij}] - ([G_4]^{-1}[G_i])^T [\bar{M}_{j4}]^T - [\bar{M}_{i4}][G_4]^{-1}[G_j] \\ &+ ([G_4]^{-1}[G_i])^T [\bar{M}_{44}][G_4]^{-1}[G_j] \quad \begin{matrix} i = 1, 2, 3 \\ j = 1, 2, 3 \end{matrix} \end{aligned} \quad (151)$$

Notice that if  $i$  and  $j$  were interchanged the matrix would be transposed; this indicates that the whole mass matrix is symmetric.

4. Using the locations of the three grid points in basic coordinates, calculate the  $3 \times 3$  transformation matrices  $[T_j]$ ,  $j = 1, 2, 3$ . Form the  $6 \times 6$  matrices:

$$[C_j] = \begin{bmatrix} T_j & 0 \\ 0 & T_j \end{bmatrix} \quad . \quad (152)$$

5. The  $3 \times 3$  mass matrix partitions are expanded to  $6 \times 6$  size and transformed to global coordinates using the logic:

$$M_3 = \frac{m}{6} X_2 Y_3 \quad , \quad (153)$$

where  $m$  is the nonstructural mass, and  $X_2, Y_3$  are the  $x$ -coordinate and  $y$ -coordinate of points 2 and 3 respectively.

# MODULE FUNCTIONAL DESCRIPTIONS

$$[\bar{M}_{ij}^a] = \begin{bmatrix} M_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & M_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & M_{11} & M_{12} & M_{13} & 0 \\ 0 & 0 & M_{21} & M_{22} & M_{23} & 0 \\ 0 & 0 & M_{31} & M_{32} & M_{33} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (154)$$

for  $i = j = 1, 2$  or  $3$ ; and

$$[\bar{M}_{ij}^e] = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & M_{11} & M_{12} & M_{13} & 0 \\ 0 & 0 & M_{21} & M_{22} & M_{23} & 0 \\ 0 & 0 & M_{31} & M_{32} & M_{33} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (155)$$

for  $i \neq j$ .

The mass matrix partitions in global coordinates are:

$$[M_{ij}^g] = [C_i]^T [E] [\bar{M}_{ij}^e] [E]^T [C_j] \quad , \quad (156)$$

where

$$[C_i] = \begin{bmatrix} T_i & | & 0 \\ \hline 0 & | & T_i \end{bmatrix} \quad (157)$$



is the global-to-basic transformation matrix,

and

$$[E] = \begin{bmatrix} \{i\} & \{j\} & \{k\} & 0 & 0 & 0 \\ 0 & 0 & 0 & \{i\} & \{j\} & \{k\} \end{bmatrix} \quad (158)$$

is the 6 x 6 element-to-basic transformation matrix.

#### 4.87.5.13 Mass Matrix Calculations for the QDPLT Element (Subroutine MQDPLT of Module SMA2).

1. The general calculations for the mass matrix of the quadrilateral plate element, QDPLT, are the same as those for the stiffness matrix calculations (see Equations 101 through 123).
2. For each subtriangle, the output from the basic bending triangle subroutine are the nine 3 x 3 matrices:

$$\begin{bmatrix} M_{aa} & M_{ab} & M_{ac} \\ M_{ba} & M_{bb} & M_{bc} \\ M_{ca} & M_{cb} & M_{cc} \end{bmatrix}$$

These are transformed to the quadrilateral system by the following equation:

$$[\bar{M}_{ij}] = \frac{1}{2} [T]^T [M_{ij}] [T] , \quad (159)$$

where  $[T]$  is given in Equation 121.

These matrices are added to their corresponding positions in the quadrilateral matrix partitions  $[M_{ij}^e]$ , in the same manner as that for the stiffness matrix partitions of the quadrilateral (See step 3 of Section 4.87.5.8).

3. For each pivot point  $i$ , the following 3 x 3 partitions are formed:

$$[M_{ij}^e], \text{ for } j = 1, 2, 3, 4 .$$

4. The mass at each point in the plane of the element is due to the mass of the attached triangles. The masses of the triangles are:

# MODULE FUNCTIONAL DESCRIPTIONS

$$m^{\beta} = \frac{m}{4} x_b^{\beta} y_c^{\beta} , \quad \beta = I, II, III \text{ and } IV , \quad (160)$$

where  $m$  is the nonstructural mass, and  $x_b^{\beta}$  and  $y_c^{\beta}$  are the x-coordinate and y-coordinate of points  $b$  and  $c$  respectively of subtriangle  $\beta$ .

The masses at the points are:

$$m_1 = \frac{1}{3} [m^I + m^{II} + m^{IV}] , \quad (161)$$

$$m_2 = \frac{1}{3} [m^I + m^{II} + m^{III}] , \quad (162)$$

$$m_3 = \frac{1}{3} [m^{II} + m^{III} + m^{IV}] , \quad (163)$$

$$m_4 = \frac{1}{3} [m^I + m^{III} + m^{IV}] . \quad (164)$$

5. The  $3 \times 3$  mass matrix partitions are expanded to  $6 \times 6$  matrices and the in-plane mass effects are added using the logic:

$$m_z^i = \frac{m_i h}{2} , \quad (165)$$

$$I_z^i = \frac{m_i h^2}{4} , \quad (166)$$

# MODULE FUNCTIONAL DESCRIPTIONS

$$[\bar{M}_{ij}^e] = \begin{bmatrix} m_i & 0 & 0 & 0 & -m_z^i & 0 \\ 0 & m_i & 0 & m_z^i & 0 & 0 \\ 0 & 0 & M_{11} & M_{12} & M_{13} & 0 \\ 0 & m_z^i & M_{21} & M_{22} + I_z^i & M_{23} & 0 \\ -m_z^i & 0 & M_{31} & M_{32} & M_{33} + I_z^i & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (167)$$

for  $i = j = 1$  or  $3$ ; and

$$[\bar{M}_{ij}^e] = \begin{bmatrix} m_i & 0 & 0 & 0 & m_z^i & 0 \\ 0 & m_i & 0 & -m_z^i & 0 & 0 \\ 0 & 0 & M_{11} & M_{12} & M_{13} & 0 \\ 0 & -m_z^i & M_{21} & M_{22} + I_z^i & M_{23} & 0 \\ m_z^i & 0 & M_{31} & M_{32} & M_{33} + I_z^i & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (168)$$

for  $i = j = 2$  or  $4$ ; and



# MODULE FUNCTIONAL DESCRIPTIONS

$$[\bar{M}_{ij}^e] = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & M_{11} & M_{12} & M_{13} & 0 \\ 0 & 0 & M_{21} & M_{22} & M_{23} & 0 \\ 0 & 0 & M_{31} & M_{32} & M_{33} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (169)$$

for  $i \neq j$ .

6. Using the geometry data, the 3 x 3 global-to-basic transformations  $[T_j]$  are formed for  $j = 1, 2, 3, 4$ . These are expanded to 6 x 6 matrices:

$$[C_j] = \left[ \begin{array}{c|c} T_j & 0 \\ \hline 0 & T_j \end{array} \right]. \quad (170)$$

7. The 6 x 6 element-to-basic transformation matrix is:

$$[E] = \left[ \begin{array}{c|c|c|c|c|c} \{i\} & \{j\} & \{k\} & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & \{i\} & \{j\} & \{k\} \end{array} \right]. \quad (171)$$

8. The 6 x 6 matrices are transformed to global coordinates using the equation:

$$[M_{ij}^g] = [C_i]^T [E] [\bar{M}_{ij}^e] [E]^T [C_j] . \quad (172)$$

# MODULE FUNCTIONAL DESCRIPTIONS

## 4.87.5.14 Thermal Load Equations for the Bending Elements (Subroutines TRBSC, TRPLT, and ODPLT of Module SSG1).

1. The phase I SDR2 routines for all bending elements generate for connected points  $g_1, g_2, g_3$ , (and  $g_4$ ) the 5x6 matrices  $[S_1], [S_2], [S_3]$  (and  $[S_4]$ ). These matrices are generated in the SSG1 module with a minor change in the basic triangle routine.
2. The matrix  $[K_s]$  is described on page 4.87-85. In the SDR2 routine the values  $x$  and  $y$  depend on which element is actually being used. For SSG1 module, calculate the matrix:

$$[K_s^t] = A[K_s(\bar{x}, \bar{y})] \quad , \quad (173)$$

the definition of  $[K_s]$  is:

$$[K_s] = \begin{bmatrix} DH_{\chi q} \\ \hline G_2 H_{\gamma q} \end{bmatrix} \quad , \quad (174)$$

where  $A$  is the area of the basic triangle and  $(\bar{x}, \bar{y})$  is the center location of the basic triangle. The lower partition will not be used.

3. For each type of element the vector,  $\{\chi\}$ , is generated where:

$$\{\chi_e\} = - \begin{Bmatrix} \alpha_1 T' \\ \alpha_2 T' \\ \alpha_{12} T' \\ 0 \\ 0 \end{Bmatrix} \quad \text{or} \quad \{\chi_e\} = \begin{Bmatrix} [D]^{-1} \{M_e\} \\ \hline 0 \\ 0 \end{Bmatrix} \quad , \quad (175)$$

where  $T'$  is given on a TEMPP1 data card or calculated from a TEMPP3 card,  $\alpha_1, \alpha_2, \alpha_{12}$  are the material thermal coefficient vector components,  $[D]$  is the 3 by 3 material matrix, and  $\{M_e\}$  are the thermal moments given on a TEMPP2 card.

# MODULE FUNCTIONAL DESCRIPTIONS

4. The 6x1 thermal load vectors in basic coordinates are:

$$\{P_j\} = N[S_j]^T \{x_e\} \quad , \quad j = 1, 2, 3 \text{ (and 4)} \quad (176)$$

where N = 1: TRBSC

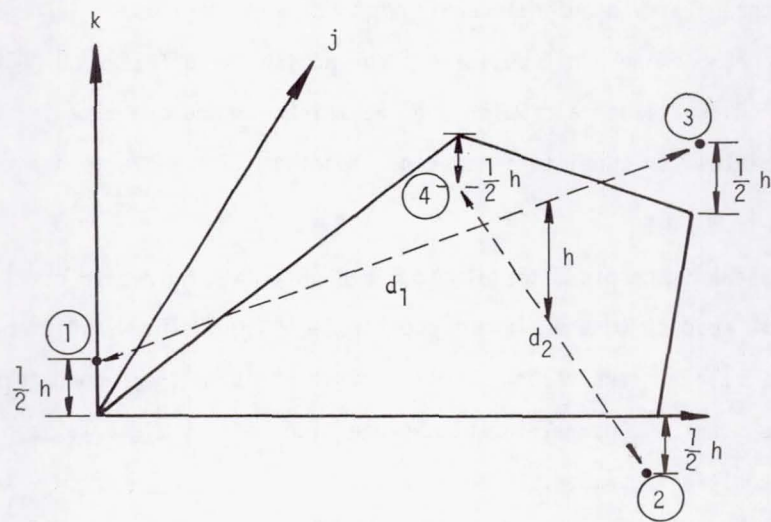
N = 3: TRPLT, TRIA1, TRIA2

N = 2: QDPLT, QUAD1, QUAD2

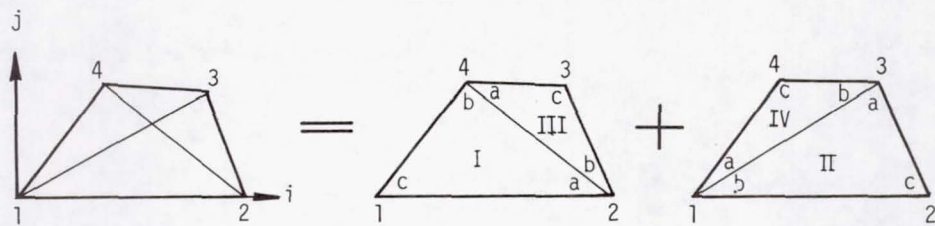
The  $[S_j]^T$  matrix is calculated from the  $[K_s]$  matrices of the various basic triangles forming the element. These  $[K_s]$  matrices are transformed and combined to produce relations  $[S_j]$  between the average element moments and the displacements of the connected grid points, j.



# STRUCTURAL ELEMENT DESCRIPTIONS



a) Definition of plane



b) Subtriangles

Figure 6. Geometry of the quadrilateral plate element, QDPLT.

## MODULE FUNCTIONAL DESCRIPTIONS

### 4.87.6 The TRIA1, TRIA2, QUAD1 and QUAD2 Elements

These elements have the properties of both membranes and bending plates. The TRIA1 and QUAD1 elements are triangles and quadrilaterals which may use separate thicknesses and materials for membrane, bending and transverse shear action. The TRIA2 and QUAD2 elements are triangles and quadrilaterals which use one thickness and one material to simulate a homogeneous plate with consistent membrane, bending and transverse shear properties.

If these elements use anisotropic materials (defined on a MAT2 bulk data card), the material is oriented with respect to the element coordinate system. The definition of the coordinate system is as follows: the vector from the first point to the second point defines the base or x axis. The z axis is normal to the average plane of the elements, and the third and fourth points have positive y values.

Mass matrices, thermal loads, and differential stiffness matrices for these elements use only the membrane properties.

#### 4.87.6.1 Input Data for the TRIA1, TRIA2, QUAD1 and QUAD2 Elements

1. The ECPT/EST entries for the TRIA1 or QUAD1 elements are:

| <u>Symbol</u>  | <u>Description</u>   |
|--|--|
| $SIL_i, \quad \left. \begin{array}{l} i = 1,2,3 \\ \text{or } i = 1,2,3,4 \end{array} \right\}$  | Scalar indices of the connected grid points.   |
| $\left. \begin{array}{l} N_i \\ X_i \\ Y_i \\ Z_i \end{array} \right\} \quad \left. \begin{array}{l} i = 1,2,3 \text{ or} \\ i = 1,2,3,4 \end{array} \right\}$ | Referenced local coordinate system and location in basic coordinates of connected grid points. |
| $\theta$   | Material orientation angle.  |
| MAT ID <sub>m</sub>  | Material identification number for membrane properties.  |
| $t_m$  | Membrane thickness.  |
| MAT ID <sub>b</sub>  | Material identification number for bending properties.   |

## STRUCTURAL ELEMENT DESCRIPTIONS

| <u>Symbol</u> | <u>Description</u>  |
|---------------|---|
| $I$           | Bending inertia   |
| $MAT ID_s$    | Material identification number for transverse shear properties. |
| $t_s$         | Transverse shear thickness                                      |
| $\mu$         | Nonstructural mass per area                                     |
| $Z_1, Z_2$    | Outer fiber distances for stress calculations.                  |
| $t_\mu$       | Temperature for material properties                             |

### 2. ECPT Entries for the TRIA2 or QUAD2 Elements.

| <u>Symbol</u>   | <u>Description</u>  |
|---|---|
| $SIL_i \quad \begin{matrix} i = 1,2,3 \text{ or} \\ i = 1,2,3,4 \end{matrix}$   | Scalar indices of connected grid points.  |
| $\left. \begin{matrix} N_i \\ X_i \\ Y_i \\ Z_i \end{matrix} \right\} \begin{matrix} i = 1,2,3, \text{ or} \\ i = 1,2,3,4 \end{matrix}$ | Referenced local coordinate system and location vector in basic coordinates of connected grid points. |
| $\theta$  | Material orientation angle.   |
| $MAT ID$  | Material identification number  |
| $t_m$   | Element thickness   |
| $\mu$   | Nonstructural mass per area.  |
| $t_\mu$   | Temperature for material properties   |

### 4.87.6.2 Stiffness Matrix Calculations (Subroutine KTTRIQD of Module SMA1).

The TRIA1 or QUAD1 element ECPT data are rearranged to correspond to the (TRMEM or QDMEM) membrane ECPT form, and the routine of the TRMEM or QDMEM element is used. The ECPT data are then rearranged to correspond to the ECPT data of a TRPLT or QDPLT element and the respective plate routine is used. Each routine is entirely independent.



## MODULE FUNCTIONAL DESCRIPTIONS

The TRIA2 and QUAD2 elements are treated in the same manner except that the arrangement of the ECPT data is different. The type "2" element uses the single material for all three material properties of the type "1" element. The membrane and transverse shear thickness equal the single thickness of the type "2" element. The bending inertia,  $I_b$ , for the plate property is:

$$I_b = \frac{t^3}{12} \quad (1)$$

### 4.87.6.3 Lumped Mass Matrix Generation (Subroutine MASSTQ of Module SMA2)

The bending properties are disregarded for the lumped mass matrix calculations, and the element mass matrices are computed exactly as the ones for the TRMEM and QDMEM elements.

### 4.87.6.4 Thermal Load Calculations (Subroutine of Module SSG1)

The TRPLT and QDPLT element routines are used to generate loads due to thermal gradients or moments. The TRMEM and QDMEM routines are used to calculate in-plane loads due to uniform thermal expansion.

### 4.87.6.5 Element Stress and Force Calculations (Subroutines STRQD1 and STRQD2 of Module SDR2)

As with the stiffness matrix calculations, the data are rearranged and the stresses for both the membrane and plate deformations are calculated. The element forces are calculated for the plate only.

1. For the TRIA2 and QUAD2 elements the outer fiber distances  $Z_1$  and  $Z_2$  are:

$$Z_1 = \frac{t}{2} \quad (2)$$

$$Z_2 = -\frac{t}{2} \quad (3)$$

The membrane and plate stresses are added together as follows for  $Z_1$ :

$$\begin{pmatrix} \sigma_{x1} \\ \sigma_{y1} \\ \sigma_{xy1} \end{pmatrix}_{\text{composite}} = \begin{pmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{pmatrix}_{\text{membrane}} + \begin{pmatrix} \sigma_{x1} \\ \sigma_{y1} \\ \sigma_{xy1} \end{pmatrix}_{\text{bending}}, \quad (4)$$

## STRUCTURAL ELEMENT DESCRIPTIONS

and for  $Z_2$ ,

$$\begin{Bmatrix} \sigma_{x2} \\ \sigma_{y2} \\ \sigma_{xy2} \end{Bmatrix}_{\text{composite}} = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{Bmatrix}_{\text{membrane}} + \begin{Bmatrix} \sigma_{x2} \\ \sigma_{y2} \\ \sigma_{xy2} \end{Bmatrix}_{\text{bending}} \quad (5)$$

2. The principal stresses and their orientation are calculated from the above results, as in Equations 28, 29 and 30 of section 4.87.4.6.

### 4.87.6.6 Coupled Mass Matrix Calculations (Subroutine MTRIQD of Module SMA2)

In the lumped mass case these elements are processed using the membrane mass calculation routines (subroutine MASSTQ of module SMA2). When coupled mass is requested, the plate mass calculations will be used instead. The ECPT data are rearranged to the appropriate TRPLT or QDPLT format, and the respective plate routine is used. The mass per area is now calculated using the material mass density  $\rho$  and the thickness  $t_m$  for the membrane definition of the element and added to the nonstructural mass:

$$m = \mu + \rho t_m, \quad (6)$$

and  $m$  is now used instead of  $\mu$  for the plate calculations.



## MODULE FUNCTIONAL DESCRIPTIONS

### 4.87.6.7 Piecewise Linear Analysis Calculations (Subroutines PSTRI1, PSTRI2, PSQAD1, and PSQAD2 of Module PLA3 and PKTRI1, PKTRI2, PKQAD1 and PKQAD2 of Module PLA4).

The additional ECPTNL and ESTNL entries are:

$\epsilon_0^*$  - The previously computed strain value once removed.

$\epsilon^*$  - The previously computed strain value.

$E^*$  - The previously computed modulus of elasticity.

$\sigma_x^*$   
 $\sigma_y^*$   
 $\sigma_{xy}^*$

} The previously computed membrane stresses.

$M_x^*$   
 $M_y^*$   
 $M_{xy}^*$

} The previously computed element forces (ESTNL only).

$V_x^*$   
 $V_y^*$

All of the above values are initially zero with the exception of  $E^*$ , which is initially the original modulus of elasticity present on a MAT1 card.

In module PLA3, the incremental element stress matrix is calculated by first rearranging the ESTNL data to correspond to the ESTNL data for a TRMEM or QDMEM, and then the membrane stresses are calculated in the same manner as Equations 103 through 105 of section 4.87.4.14. Then the ESTNL data are rearranged to correspond to the EST data for a TRPLT (or QDPLT) and the incremental bending forces for the TRPLT (or QDPLT) element are calculated in the same manner as in Equation 97 of section 4.87.5.7. However, if the bending material properties are the same as the membrane material properties, then the 3 by 3 bending material properties matrix ( $[G_b]$  in Equation 11 of section 4.87.5.2) is replaced by the matrix given in Equation 97 of section 4.87.4.14. In addition the displacement vector  $\{u_i\}$  in Equation 44 of section



# STRUCTURAL ELEMENT DESCRIPTIONS

4.87.5.4 (or Equation 97 of section 4.87.5.7) are replaced by an incremental displacement vector  $\{\Delta u_i\}$ .

The results are incremental stresses and forces for the membrane and bending properties defined as follows

$$\sigma_{x1} = \sigma_x^* + \Delta\sigma_x \quad (7)$$

$$\sigma_{y1} = \sigma_y^* + \Delta\sigma_y \quad (8)$$

$$\sigma_{xy1} = \sigma_{xy}^* + \Delta\sigma_{xy} \quad (9)$$

Membrane stresses ,

$$M_{x1} = M_x^* + \Delta M_x \quad (10)$$

$$M_{y1} = M_y^* + \Delta M_y \quad (11)$$

$$M_{xy1} = M_{xy}^* + \Delta M_{xy} \quad (12)$$

Bending forces .

$$V_{x1} = V_x^* + \Delta V_x \quad (13)$$

$$V_{y1} = V_y^* + \Delta V_y \quad (14)$$

The total bending stresses are calculated using the total bending forces given in Equations 10 through 14 in conjunction with Equations 98 through 100 of section 4.87.5.7.

The membrane and bending stresses are added together as follows for  $Z_1$ :

$$\begin{Bmatrix} \sigma_{x1} \\ \sigma_{y1} \\ \sigma_{xy1} \end{Bmatrix}_{\text{composite}} = \begin{Bmatrix} \sigma_{x1} \\ \sigma_{y1} \\ \sigma_{xy1} \end{Bmatrix}_{\text{membrane}} + \begin{Bmatrix} \sigma_{x1} \\ \sigma_{y1} \\ \sigma_{xy1} \end{Bmatrix}_{\text{bending}} ; \quad (15)$$

# STRUCTURAL ELEMENT DESCRIPTIONS

and for  $Z_2$ :

$$\begin{Bmatrix} \sigma_{x2} \\ \sigma_{y2} \\ \sigma_{xy2} \end{Bmatrix}_{\text{composite}} = \begin{Bmatrix} \sigma_{x1} \\ \sigma_{y1} \\ \sigma_{xy1} \end{Bmatrix}_{\text{membrane}} + \begin{Bmatrix} \sigma_{x2} \\ \sigma_{y2} \\ \sigma_{xy2} \end{Bmatrix}_{\text{bending}} \quad (16)$$

The principal stresses and their orientation for output are calculated from the above results, as in Equations 28 through 30 of section 4.87.4.6.

In module PLA4, the bending properties are disregarded, and the ECPTNL data are rearranged to correspond to the ECPT data for the TRMEM or QDMEM, and the stresses are calculated exactly as the ones for the TRMEM or QDMEM elements (see section 4.87.4.14).

In modules PLA3 and PLA4, after the above stress calculations have been completed, the next elastic coefficients are calculated in the same manner as Equations 106 through 112 of section 4.87.4.14.

The new ESTNL and ECPTNL entries are:

$$\epsilon_0^* = \epsilon^* \quad , \quad (17)$$

$$\epsilon^* = \epsilon_1^* \quad , \quad (18)$$

$$E^* = E_1^* \quad , \quad (19)$$

$$\sigma_x^* = \sigma_{x1}^* \quad , \quad (20)$$

$$\sigma_y^* = \sigma_{y1}^* \quad , \quad (21)$$

$$\sigma_{xy}^* = \sigma_{xy1}^* \quad , \quad (22)$$

$$M_x^* = M_{x1}^* \quad , \quad (23)$$

$$M_y^* = M_{y1}^* \quad , \quad (24)$$

$$M_{xy}^* = M_{xy1}^* \quad , \quad (25)$$

## STRUCTURAL ELEMENT DESCRIPTIONS

$$V_{x1}^* = V_{x1} \quad , \quad (26)$$

$$V_{y1}^* = V_{y1} \quad , \quad (27)$$

In module PLA4, the ECPTNL data are rearranged, and the element stiffness matrices are calculated in the same manner as for the TRMEM or QDMEM elements in section 4.87.4.14.

### 4.87.6.8 Differential Stiffness Matrix Calculations for the TRIA1 and TRIA2 Elements

(Subroutine DTRIA of Module DSMG1)

This subroutine uses the displacement vectors and thermal load temperature from a static solution to produce a differential stiffness matrix. The DTRMEM subroutine is used to calculate in plane-stresses and in-plane differential stiffness. The triangle is subdivided into three subtriangles and the DTRBSC routine is used to calculate out-of-plane differential stiffness for each of the three subtriangles. The matrices are combined and the center point is removed in the same manner as the KTRPLT subroutine (section 4.87.5.5). The basic steps are as follows:

1. The TRIA2 data from the ECPT is converted to TRIA1 format while in both cases the data is moved to a protected location. The conversion is:

$$\begin{array}{ccccc} \text{TRIA1 DATA} & & \text{TRIA2 DATA} & & \\ \text{MATID}_m = \text{MATID}_b = \text{MATID}_s = \text{MATID} & & & & (28) \end{array}$$

$$t_m = t_s = t_m \quad (29)$$

$$I = \frac{t_m^3}{12.0} \quad (30)$$

2. The ECPT data are rearranged to the TRMEM format (the same as the TRIA2 format).
3. The material property orientation angles are established and subroutine DTRMEM is called. This routine will insert the in-plane differential stiffness terms in the KDGG matrix and will return the stress values  $\bar{\sigma}_x$ ,  $\bar{\sigma}_y$ , and  $\bar{\tau}_{xy}$ .
4. The element coordinate system and geometric coefficients are calculated as follows where the three location vectors are  $\{x(1)\}$ ,  $\{x(2)\}$ , and  $\{x(3)\}$ .

$$\{V_2\} = \{x(2)\} - \{x(1)\} \quad (31)$$

$$\{V_3\} = \{x(3)\} - \{x(1)\} \quad (32)$$



# MODULE FUNCTIONAL DESCRIPTIONS

$$X_2 = |\{V_2\}| \quad (33)$$

$$\{i\} = \frac{\{V_2\}}{X_2} \quad (34)$$

$$Y_3 = |\{i\} \times \{V_3\}| \quad (35)$$

$$\{k\} = \frac{\{i\} \times \{V_3\}}{Y_3} \quad (36)$$

$$\{j\} = \{k\} \times \{i\} \quad (37)$$

$$X_4 = \frac{X_2 + X_3}{3} \quad (38)$$

$$Y_4 = \frac{Y_3}{3} \quad (39)$$

The locations of the four points in this coordinate system are defined by the matrix [R] where each row defines one point

$$[R] = \begin{bmatrix} 0 & 0 \\ X_2 & 0 \\ X_3 & Y_3 \\ X_4 & Y_4 \end{bmatrix} \quad (40)$$

5. For use in transferring points in the element to points in the subtriangle, the integer mapping matrix [M] is used.

$$[M] = \begin{array}{ccc} & \text{Point a} & \text{Point b} & \text{Point c} \\ \begin{bmatrix} 1 & 2 & 4 \\ 2 & 3 & 4 \\ 3 & 1 & 4 \end{bmatrix} & & & \end{array} \quad (41)$$

Each row of the matrix corresponds to the three points connected to each subtriangle.

6. A major loop is now performed with one cycle for each of the three subtriangles. Corresponding to points a, b, and c of the M matrix, the location vectors  $\{r_a\}$ ,  $\{r_b\}$ , and  $\{r_c\}$  are extracted from the corresponding rows of the R matrix. For each triangle the

# STRUCTURAL ELEMENT DESCRIPTIONS

following are calculated:

$$\ell = \sqrt{(r_{b1} - r_{a1})^2 - (r_{b2} - r_{a2})^2} \quad (42)$$

$$W_1 = \frac{1}{\ell} (r_{b1} - r_{a1}) \quad (43)$$

$$W_2 = \frac{1}{\ell} (r_{b2} - r_{a2}) \quad (44)$$

The transformation between subtriangle coordinates to element coordinates is:

$$[T] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & W_1 & W_2 \\ 0 & -W_2 & W_1 \end{bmatrix} \quad (45)$$

The material orientation data are

$$\left. \begin{aligned} \sin(\theta_m) &= W_1 \sin(\theta) - W_2 \cos(\theta) \\ \cos(\theta_m) &= W_1 \cos(\theta) + W_2 \sin(\theta) \end{aligned} \right\} \quad (46)$$

7. The locations of the three points of each subtriangle are transformed to geometric coefficients  $x_b, x_c, y_c$  where

$$x_b = W_1(r_{b1} - r_{a1}) + W_2(r_{b2} - r_{a2}) \quad (47)$$

$$x_c = W_1(r_{c1} - r_{a1}) + W_2(r_{c2} - r_{a2}) \quad (48)$$

$$y_c = -W_2(r_{c1} - r_{a1}) + W_1(r_{c2} - r_{a2}) \quad (49)$$

8. The stresses are transformed to the subtriangle system by the equations:

$$\sigma_x = W_1^2 \bar{\sigma}_x + W_2^2 \bar{\sigma}_y + 2W_1W_2 \bar{\tau}_{xy} \quad (50)$$

$$\sigma_y = W_2^2 \bar{\sigma}_x + W_1^2 \bar{\sigma}_y - 2W_1W_2 \bar{\tau}_{xy} \quad (51)$$

$$\tau_{xy} = -W_1W_2 \bar{\sigma}_x + W_1W_2 \bar{\sigma}_y + (W_1^2 - W_2^2) \bar{\tau}_{xy} \quad (52)$$

## MODULE FUNCTIONAL DESCRIPTIONS

9. The differential stiffness matrix for each subtriangle is formed in subroutine DTRBSC (Section 4.87.6.10). The input to this routine consists of the ECPT property data; the location parameters  $x_b$ ,  $x_c$ , and  $y_c$ ; and the in plane stresses  $\sigma_x$ ,  $\sigma_y$  and  $\tau_{xy}$ . The output from this routine consists of the following matrices:

$$[K_{ij}^d] \quad \begin{cases} i = \text{pivot point} \\ j = 1, 2, \text{ and/or } 3 \end{cases}$$

$$[K_{j4}^d] \quad j = 1, 2, \text{ and/or } 3$$

$$[H], [S]$$

10. The above matrices are combined and transformed in exactly the same manner as the triangular plate equations. The differential stiffness matrices replace the elastic stiffness matrices in the equations. See section 4.87.5.5, steps 7 through 10.

### 4.87.6.9 Differential Stiffness Matrix Calculation for the QUAD1 and QUAD2 Elements (Subroutine DQUAD of Module DSMG1)

The differential stiffness matrix for the QUAD1 and QUAD2 elements is constructed from the matrices produced by four subtriangles. The method used to subdivide the quadrilateral is shown in Figure 6. The stress is calculated for each triangle using the DTRMEM subroutine. The out-of-plane differential stiffness for each triangle is calculated using the DTRBSC subroutine. The element geometry and the manipulation of the matrices is done in the same manner as the elastic stiffness equations for the quadrilateral plates.

The steps followed by the subroutine are:

1. If a QUAD2 element is used, its property data is converted to the QUAD1 equivalent and the ECPT is expanded to the QUAD1 format. The property conversion is:

$$\text{MATID}_m = \text{MATID}_6 = \text{MATID}_5 = \text{MATID} \quad (53)$$

$$t_m = t_s = t_m \quad (54)$$

$$I = \frac{t_m^3}{12.0} \quad (55)$$



# STRUCTURAL ELEMENT DESCRIPTIONS

With both cases the data is stored in a protected location.

2. The element coordinate system and the location of the grid points are calculated as follows:

$$\left. \begin{aligned} \{d_1\} &= \{V_3\} - \{V_1\} \\ \{d_2\} &= \{V_4\} - \{V_2\} \end{aligned} \right\} \quad (56)$$

where  $\{V_1\}$ ,  $\{V_2\}$ ,  $\{V_3\}$ , and  $\{V_4\}$  are the location vectors of the connected grid points.

$$\{k\} = \frac{\{d_1\} \times \{d_2\}}{|\{d_1\} \times \{d_2\}|} \quad (57)$$

$$\{a_1\} = \{V_2\} - \{V_1\} \quad (58)$$

$$h = \{a_1\} \cdot \{k\} \quad (59)$$

$$\{i\} = \frac{\{a_1\} - h\{k\}}{|\{a_1\} - h\{k\}|} \quad (60)$$

$$\{j\} = \{k\} \times \{i\} \quad (61)$$

The locations of the points in the element plane are stored in the [R] matrix where each row corresponds to the x and y location of a point.

$$R_{11} = R_{12} = R_{22} = 0.0 \quad (62)$$

$$R_{21} = X_2 = \{a_1\} \cdot \{i\} \quad (63)$$

$$R_{31} = X_3 = \{d_1\} \cdot \{i\} \quad (64)$$

$$R_{32} = Y_3 = \{d_1\} \cdot \{j\} \quad (65)$$

$$R_{41} = X_4 = X_2 + \{d_2\} \cdot \{i\} \quad (66)$$

$$R_{42} = Y_4 = \{d_2\} \cdot \{j\} \quad (67)$$

3. At this stage the four triangles are processed in a loop. The matrix partition for the element is set to zero and the results for each triangle are added in. The following steps 4 - 8 describe this loop.

# MODULE FUNCTIONAL DESCRIPTIONS

4. The three points for each triangle are selected using the mapping matrix M. The data corresponding to these points are put into the ECPT format for a TRMEM element.
5. The geometry of the triangle is calculated using the three rows of the R matrix corresponding to the three points of the triangle a, b, c. These rows correspond to vectors  $\{V_a\}$ ,  $\{V_b\}$ , and  $\{V_c\}$ .

$$\{V\} = \{V_b\} - \{V_a\} \quad (68)$$

$$\{V_v\} = \{V_c\} - \{V_a\} \quad (69)$$

$$x_b = |\{V\}| \quad (70)$$

$$\begin{Bmatrix} w_1 \\ w_2 \end{Bmatrix} = \frac{\{V\}}{x_b} \quad (71)$$

$$x_c = w_1 V_{v1} + w_2 V_{v2} \quad (72)$$

$$y_c = -w_2 V_{v1} + w_1 V_{v2} \quad (73)$$

The transformation matrix between element and triangle coordinates is:

$$[T] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & w_1 & w_2 \\ 0 & -w_2 & w_1 \end{bmatrix} \quad (74)$$

6. The orientation angle  $\theta_m$  for the subtriangle is computed in case the material is anisotropic.

$$\sin \theta_m = w_1 \sin \theta - w_2 \cos \theta \quad (75)$$

$$\cos \theta_m = w_1 \cos \theta + w_2 \sin \theta \quad (76)$$

7. The triangular membrane subroutine DTRMEM at this point will calculate and insert the in-plane differential stiffness terms and will produce the stress values  $\sigma_x$ ,  $\sigma_y$ , and  $\tau_{xy}$  in the triangle coordinates. The basic triangle subroutine will use the in-plane stresses and the basic locations  $x_b$ ,  $x_c$ , and  $y_c$  to produce the out of plane differential stiffness terms. The output of the differential stiffness subroutine, DTRBSC, is the 3 by 3 matrix

## STRUCTURAL ELEMENT DESCRIPTIONS

partitions  $[K_{ia}]$ ,  $[K_{ib}]$ , and  $[K_{ic}]$ , where  $i$  is the pivot point. These are transformed to quadrilateral coordinates with the  $[T]$  matrix.

$$[K_{ij}^e] = [T]^T [K_{ij}] [T] \quad (77)$$

8. The matrices for each triangle are added into the running sum for the quadrilateral and steps 4 - 8 are repeated.

9. The differential stiffness matrix partitions are transformed to global coordinates and inserted into the overall differential stiffness matrix in a manner identical to steps 7 and 8 of section 4.87.5.8.

### 4.87.6.10 Differential Stiffness Matrix Calculations for the Basic Bending Triangle (Subroutine DTRBSC of Module DSMG1)

Unlike the case of elastic stiffness matrix generation, the basic triangle (TRBSC) may not be used by itself to produce differential stiffness matrix terms. This subroutine, however, is used for the calculation of differential stiffness for the TRIA1, TRIA2, QUAD1, and QUAD2 elements. Its purpose is analogous to the way the KTRBSC subroutine is used in the calculation of elastic stiffness matrices.

The necessary inputs to this subroutine are passed to it via the labeled common blocks DSIAET and DSIADP. The input data used are  $x_b$ ,  $x_c$ ,  $y_c$  (the basic geometry),  $\sigma_x$ ,  $\sigma_y$ ,  $\tau_{xy}$  (the in-plane stresses), and the element property data.

The basic algorithm used by the routine is as follows:

1. The presence of transverse shear is tested and the subroutine selects the method of calculating the element coordinate-to-generalized coordinate transformation matrices  $[H_b]$  and  $[H_c]$ .
2. If no transverse shear flexibility exists, the matrices  $[H_b]$  and  $[H_c]$  are calculated by the following equations:



# MODULE FUNCTIONAL DESCRIPTIONS

$$\begin{aligned} r &= \frac{1}{x_b} \\ s &= \frac{1}{y_c} \\ t &= \frac{x_c}{y_c} \end{aligned} \tag{78}$$

$$u = \frac{x_c}{x_b^2 y_c^2} = r^2 s t$$

$$[H_b] = \begin{bmatrix} 3r^2 & 0 & r \\ 0 & r & 0 \\ -3r^2 t^2 & -rt & -rt^2 \\ -2r^3 & 0 & -r^2 \\ -6ru(x_b - x_c) & -rs & u(3x_c - 2x_b) \\ 2rtu(3x_b - 2x_c) & rst & 2tu(x_b - x_c) \end{bmatrix} \quad (6 \times 3) \tag{79}$$

$$[H_c] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 3s^2 & -s & st \\ 0 & 0 & 0 \\ 0 & 0 & -s^2 \\ -2s^3 & s^2 & 0 \end{bmatrix} \quad (6 \times 3) \tag{80}$$

3. If both shear material and shear thickness exist, then the  $[H_{\gamma q}]$  and  $[H]^{-1}$  matrices are generated as with the existing equations for the TRBSC element. See pages 4.87-82, 83. The  $6 \times 6$   $[H]^{-1}$  matrix, is partitioned as follows:

$$[H]^{-1} \Rightarrow [H_b : H_c] \tag{81}$$

and is used instead of Equations (79) and (80).

# STRUCTURAL ELEMENT DESCRIPTIONS

4. In order to form the differential stiffness matrix  $[K^{dq}]$ , referred to generalized coordinates, the following integral must be evaluated over the triangular area.

$$I_{nm} = h \int_A x^n y^m dA \quad (82)$$

The results are:

$$\begin{aligned} I_{00} &= hA = \frac{hx_b y_c}{2} & I_{10} &= \frac{hA}{3} (x_b + x_c) \\ I_{01} &= \frac{hAy_c}{3} & I_{11} &= \frac{hAy_c}{12} (x_b + 2x_c) \\ I_{02} &= \frac{hAy_c^2}{6} & I_{12} &= \frac{hAy_c^2}{30} (x_b + 3x_c) \\ I_{03} &= \frac{hAy_c^3}{10} & I_{13} &= \frac{hAy_c^3}{60} (x_b + 4x_c) \\ I_{04} &= \frac{hAy_c^4}{15} \\ I_{20} &= \frac{hA}{6} (x_b^2 + x_b x_c + x_c^2) & (83) \\ I_{21} &= \frac{hAy_c}{30} (x_b^2 + 2x_b x_c + 3x_c^2) \\ I_{22} &= \frac{hAy_c^2}{90} (x_b^2 + 3x_b x_c + 6x_c^2) \\ I_{30} &= \frac{hA}{10} [x_b^3 + x_b^2 x_c + x_b x_c^2 + x_c^3] \\ I_{31} &= \frac{hAy_c}{60} [x_b^3 + 2x_b^2 x_c + 3x_b x_c^2 + 4x_c^3] \\ I_{40} &= \frac{hA}{15} [x_b^4 + x_b^3 x_c + x_b^2 x_c^2 + x_b x_c^3 + x_c^4] \end{aligned}$$

# MODULE FUNCTIONAL DESCRIPTIONS

5. The elements of the (8x8) differential stiffness matrix  $[K^{dq}]$  are listed below. The matrix is symmetric so only the upper triangle terms are given. The superscript  $(dq)$  is omitted for convenience.

$$\begin{aligned}
 K_{11} &= \sigma_x I_{00} \\
 K_{12} &= \tau I_{00} \\
 K_{13} &= 2\sigma_x I_{10} \\
 K_{14} &= \sigma_x I_{01} + \tau I_{10} \\
 K_{15} &= 2\tau I_{01} \\
 K_{16} &= 3\sigma_x I_{20} \\
 K_{17} &= \sigma_x I_{02} + 2\tau I_{11} \\
 K_{18} &= 3\tau I_{02} \\
 K_{22} &= \sigma_y I_{00} \\
 K_{23} &= 2\tau I_{10} \\
 K_{24} &= \tau I_{01} + \sigma_y I_{10} \\
 K_{25} &= 2\sigma_y I_{01} \\
 K_{26} &= 3\tau I_{20} \\
 K_{27} &= \tau I_{02} + 2\sigma_y I_{11} \\
 K_{28} &= 3\sigma_y I_{02} \\
 K_{33} &= 4\sigma_x I_{20} \\
 K_{34} &= 2(\sigma_x I_{11} + \tau I_{20}) \\
 K_{35} &= 4\tau I_{11} \\
 K_{36} &= 6\sigma_x I_{30}
 \end{aligned}
 \tag{84}$$



# STRUCTURAL ELEMENT DESCRIPTIONS

$$\begin{aligned}
 K_{37} &= 2(\sigma_x I_{12} + 2\tau I_{21}) \\
 K_{38} &= 6\tau I_{12} \\
 K_{44} &= \sigma_x I_{02} + 2\tau I_{11} + \sigma_y I_{20} \\
 K_{45} &= 2(\tau I_{02} + \sigma_y I_{11}) \\
 K_{46} &= 3(\sigma_x I_{21} + \tau I_{30}) \\
 K_{47} &= \sigma_x I_{03} + 3\tau I_{12} + 2\sigma_y I_{21} \\
 K_{48} &= 3(\tau I_{03} + \sigma_y I_{12}) \\
 K_{55} &= 4\sigma_y I_{02} \\
 K_{56} &= 6\tau I_{21} \\
 K_{57} &= 2(\tau I_{03} + 2\sigma_y I_{12}) \\
 K_{58} &= 6\sigma_y I_{03} \\
 K_{66} &= 9\sigma_x I_{40} \\
 K_{67} &= 3(\sigma_x I_{22} + 2\tau I_{31}) \\
 K_{68} &= 9\tau I_{22} \\
 K_{77} &= \sigma_x I_{04} + 4\tau I_{13} + 4\sigma_y I_{22} \\
 K_{78} &= 3(\tau I_{04} + 2\sigma_y I_{13}) \\
 K_{88} &= 9\sigma_y I_{04}
 \end{aligned} \tag{84}$$

6. In order to transform the matrix to the displacements of points at the corners of the triangle, the following matrices are generated.

$$[H_a] = [H_b][S_b] - [H_c][S_c] \quad (6 \times 3) \tag{85}$$

# MODULE FUNCTIONAL DESCRIPTIONS

$$[C_a] = \begin{bmatrix} H_q H_a \\ \hline H_a \end{bmatrix} + \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ \hline 0 \end{bmatrix} \quad (8 \times 3) \quad (86)$$

$$[C_b] = \begin{bmatrix} H_q H_b \\ \hline H_b \end{bmatrix} \quad (8 \times 3) \quad (87)$$

$$[C_c] = \begin{bmatrix} H_q H_c \\ \hline H_c \end{bmatrix} \quad (8 \times 3) \quad (88)$$

7. The output matrix partitions depend on the type of element using this subroutine. If the element type is a QUAD1 or QUAD2 the three output matrix partitions  $[K^{de}]$  are:

$$[K_{ij}^{de}] = [C_i]^T [K^{dq}] [C_j] \quad (89)$$

where  $i$  is the pivot point and  $j = a, b, \text{ and } c$ .

If the element type is a TRIA1 or TRIA2 the output differential stiffness matrices are calculated using equation 89 above. They are:

$$[K_{ia}^{de}] = [C_i]^T [K^{dq}] [C_a] \quad (90)$$

$$[K_{ib}^{de}] = [C_i]^T [K^{dq}] [C_b] \quad (91)$$

if  $i$  is the pivot point and  $i = a$  or  $i = b$ .

In addition, for the TRIA1 and TRIA2, elements the following matrices are output:

$$[K_{ac}^{de}] = [C_a]^T [K^{dq}] [C_c] \quad (92)$$

$$[K_{bc}^{de}] = [C_b]^T [K^{dq}] [C_c] \quad (93)$$

$$[K_{cc}^{de}] = [C_c]^T [K^{dq}] [C_c] \quad (94)$$

The matrices  $[H]^{-1}$  and  $[S]$ , previously calculated are also output for the TRIA1 and TRIA2 elements.

#### 4.87.6.11 Thermal Calculations for the Combination Elements

If the heat transfer system parameter equals +1, the elements are treated exactly like the membrane elements QDMEM and TRMEM. The bending calculations are bypassed and subroutines KQDMEM or KTRMEM are used for calculation of the conductivity matrix terms, subroutine MTRQD is used for the mass calculations, and subroutines SDHTF1, SDHTF2, and SDHTFF are used for stress recovery.



## MODULE FUNCTIONAL DESCRIPTIONS

### 4.87.7 The ELASi, MASSi and DAMPi Elements

The scalar elements (ELASi, MASSi and DAMPi,  $i = 1, 2, 3, 4$ ) are connected to scalar components of grid points or to scalar points. The ELASi elements contribute only to:

a) the stiffness matrix,  $[K_{gg}^x]$ , for  $i = 1, 2, 3, 4$ ; and b) to the structural damping matrix,  $[K_{gg}^4]$ , for  $i = 1, 2, 3$ . The MASSi elements contribute only to the mass matrix,  $[M_{gg}]$ , and the DAMPi elements contribute only to the viscous damping matrix,  $[B_{gg}]$ .

The scalar elements do not require material or geometric properties in their calculations. Only the ELASi elements are used for stress or force calculations.

#### 4.87.7.1 Input Data for the ELASi, MASSi and DAMPi Elements

The ECPT/EST entries for the scalar elements are:

| <u>Symbol</u>  | <u>Description</u>                                       | <u>Elements</u>    |
|----------------|--|--------------------|
| $SIL_1, SIL_2$ | Scalar indices of connected grid or scalar points        | All                |
| $c_1, c_2$     | Component numbers corresponding to $SIL_1$ and $SIL_2$ . | Types 1 and 2      |
| $K$            | Spring constant  | All ELASi elements |
| $g_e$          | Damping factor   | ELASi, 2 and 3     |
| $S$            | Stress coefficient                                       |                    |
| $B$            | Viscous damping coefficient                              | All DAMPi elements |
| $m$            | Mass coefficient   | All MASSi elements |

#### 4.87.7.2 ELASi Stiffness Matrix Generation (Subroutine KELAS of Module SMA1)

1. The two connected scalar indices are  $i_1$  and  $i_2$  given by:

$$i_1 = \begin{cases} SIL_1 + (c_1 - 1), & \text{if Point 1 is a grid point} \\ SIL_1, & \text{if Point 1 is a scalar point} \end{cases}$$

## STRUCTURAL ELEMENT DESCRIPTIONS

$$i_2 = \begin{cases} \text{SIL}_2 + (c_2 - 1), & \text{if Point 2 is a grid point} \\ \text{SIL}_2, & \text{if Point 2 is a scalar point} \end{cases}$$

2. The following terms are added to the  $[K_{gg}^x]$  matrix:

+K in position  $(i_1, i_1)$  and in position  $(i_2, i_2)$ ,

-K in position  $(i_2, i_1)$  and in position  $(i_1, i_2)$ .

3. If point 2 is not defined, add +K to position  $(i_1, i_1)$ .

4. The damping terms are:

$$K^4 = K g_e. \quad (1)$$

These are added to  $[K_{gg}^4]$  in the same manner as the stiffness terms were added to  $[K_{gg}^x]$ .

### 4.87.7.3 MASSi Mass Matrix Generation (Subroutine MASSD of Module SMA2)

These elements are treated like the ELASi elements except the "m" term is added to the four positions in  $[M_{gg}]$ .

### 4.87.7.4 DMAPi Damping Matrix Generation (Subroutine MASSD of Module SMA2)

These elements are treated like the ELASi elements except the "B" term is added to the four positions in  $[B_{gg}]$ .

### 4.87.7.5 ELASi Stress and Force Recovery (Subroutines SELAS1 and SELAS2 of Module SDR2)

The element force is:

$$F = K(u_2 - u_1), \quad (2)$$

where  $u_1$  and  $u_2$  are the displacements at scalar index numbers  $i_1$  and  $i_2$ .

## MODULE FUNCTIONAL DESCRIPTIONS

The element stress is:

$$\sigma = SF. \quad (3)$$



## STRUCTURAL ELEMENT DESCRIPTIONS

### 4.87.8 Concentrated Mass Elements CØNM1, CØNM2

Two types of grid point mass data are available. CØNM1 defines the mass matrix directly at the point, with the axes defined by a given local coordinate system. CØNM2 defines the same matrix for a body with a mass and three inertias with a center of gravity offset from the grid point.

#### 4.87.8.1 ECPT Entries for the CØNM1 Mass Element

| <u>Symbol</u>  | <u>Description</u>  |
|--|---|
| $N_m$  | Local coordinate system number in which the mass matrix is defined. |
| $N_g, X, Y, Z$   | Local coordinate system number and basic coordinates of the point.  |
| $m_{11}$<br>$m_{21}, m_{22}$<br>$m_{31}, m_{32}, m_{33}$<br>$m_{41}, m_{42}, m_{43}, m_{44}$<br>$m_{51}, m_{52}, m_{53}, m_{54}, m_{55}$<br>$m_{61}, m_{62}, m_{63}, m_{64}, m_{65}, m_{66}$ | Terms of mass matrix given in row form (out to the diagonal term).  |

#### 4.87.8.2 Mass Matrix Calculations for the CØNM1 Element (Subroutine MCØNMX of Module SMA2)

- Using the symmetrical relationships, fill out the remainder of the 6x6 matrix, [m]:

$$m_{ij} = m_{ji} \quad (1)$$

- Using the basic coordinates of the point and the local coordinate system definition, the 3x3 transformation matrices  $[T_g]$  and  $[T_m]$  are generated, and the mass matrix in global coordinates is:

# MODULE FUNCTIONAL DESCRIPTIONS

$$[M] = \begin{bmatrix} T_g^T & 0 \\ 0 & T_g^T \end{bmatrix} \begin{bmatrix} T_m & 0 \\ 0 & T_m \end{bmatrix} [m] \begin{bmatrix} T_m^T & 0 \\ 0 & T_m^T \end{bmatrix} \begin{bmatrix} T_g & 0 \\ 0 & T_g \end{bmatrix}, \quad (2)$$

where  $[T_g]$  is the transformation from global-to-basic coordinates at the point, and  $[T_m]$  is the transformation from the coordinates defined by the mass local system to the basic coordinate system.

## 4.87.8.3 ECPT Entries for the C0NM2 Mass Element

| <u>Symbol</u>  | <u>Description</u>   |
|--|--|
| $N_m$  | Local coordinate system number in which the mass terms are defined.              |
| $N_g, X, Y, Z$   | Local coordinate system number and basic coordinates of the point.               |
| $\bar{m}$  | Concentrated mass  |
| $x, y, z$  | Offset of center of gravity in mass coordinate system                            |
| $I_{11}$<br>$I_{21}, I_{22}$<br>$I_{31}, I_{32}, I_{33}$ | Inertias about the center of gravity given in row order out to the diagonal term |

## 4.87.8.4 Mass Matrix Calculations for the C0NM2 Element (Subroutine MC0NMX of Module SMA2)

1. The transformation from the offset to the grid point is:

$$[D] = \begin{bmatrix} 1 & & & 0 & z & -y \\ & 1 & & -z & 0 & x \\ & & 1 & y & -x & 0 \\ \hline & & & 1 & & \\ 0 & & & & 1 & \\ & & & & & 1 \end{bmatrix}. \quad (3)$$

# STRUCTURAL ELEMENT DESCRIPTIONS

2. The mass matrix referenced to the offset point is:

$$[m_0] = \begin{bmatrix} \bar{m} & & & & & \\ & \bar{m} & & & & \\ & & \bar{m} & & & \\ & & & 0 & & \\ & & & & I_{11} & -I_{21} & -I_{31} \\ & & & & -I_{21} & I_{22} & -I_{32} \\ & & & & -I_{31} & -I_{32} & I_{33} \end{bmatrix} \cdot \quad (4)$$

3. The mass matrix about the grid point along the given coordinates is:

$$[m] = [D]^T [m_0] [D]. \quad (5)$$

The actual nonzero terms of matrix  $[m]$  are calculated directly from the equations:

$$m_{11} = m_{22} = m_{33} = \bar{m}, \quad (6)$$

$$m_{15} = m_{51} = -m_{24} = -m_{42} = +z\bar{m}, \quad (7)$$

$$m_{16} = m_{61} = -m_{34} = -m_{43} = -y\bar{m}, \quad (8)$$

$$m_{26} = m_{62} = -m_{35} = -m_{53} = x\bar{m}, \quad (9)$$

$$m_{44} = I_{11} + (y^2 + z^2)\bar{m}, \quad (10)$$

$$m_{55} = I_{22} + (x^2 + z^2)\bar{m}, \quad (11)$$

$$m_{66} = I_{33} + (x^2 + y^2)\bar{m}, \quad (12)$$

$$m_{45} = m_{54} = -I_{21} - xy\bar{m}, \quad (13)$$



# MODULE FUNCTIONAL DESCRIPTIONS

$$m_{46} = m_{64} = -I_{31} - xz\bar{m} , \quad (14)$$

$$m_{56} = m_{65} = -I_{32} - yz\bar{m} . \quad (15)$$

4. The matrix  $[m]$  is transformed back to global coordinates using Equation 2.

## STRUCTURAL ELEMENT DESCRIPTIONS

### 4.87.9 The CØNEAX Element

#### 4.87.9.1 Input Data for the CØNEAX Element

1. The ECPT/EST entries for the CØNEAX element are:

| <u>Symbol</u>                    | <u>Description</u>   |
|----------------------------------|--|
| $SIL_a, SIL_b$                   | Scalar indices of the $n^{th}$ harmonic of the connected rings |
| $n$                              | Harmonic index   |
| $r_a, r_b$                       | Radii at points a and b  |
| $z_a, z_b$                       | Projected distances along the axis                             |
| $t_m$                            | Membrane thickness   |
| $MAT ID_m$                       | Membrane material identification number                        |
| $I$                              | Bending coefficient  |
| $MAT ID_b$                       | Bending material identification number                         |
| $t_s$                            | Shear thickness  |
| $MAT ID_s$                       | Shear material identification number                           |
| $Z_1, Z_2$                       | Outer fiber distances for stress calculations                  |
| $\theta_i, i = 1 \text{ to } 14$ | Angles defining points around element                          |
| $t_\mu$                          | Temperature for material properties                            |
| $\mu$                            | Nonstructural mass   |

#### 4.87.9.2 Stiffness Matrix Calculations (Subroutine KCØNE of Module SMA1).

1. The shell orientation is given by:

$$\ell = \sqrt{(r_b - r_a)^2 + (z_b - z_a)^2}, \quad (1)$$

$$\sin\psi = \frac{r_b - r_a}{\ell}, \quad (2)$$

$$\cos\psi = \frac{z_b - z_a}{\ell}. \quad (3)$$

## MODULE FUNCTIONAL DESCRIPTIONS

2. The transformation matrix  $[E]$  from element coordinates to ring cylindrical coordinates is:

$$[E] = \begin{bmatrix} 0 & \sin \psi & \cos \psi & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & \cos \psi & -\sin \psi & 0 & 0 \\ 0 & 0 & 0 & 0 & \sin \psi \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \cos \psi \end{bmatrix} \quad (4)$$

3. The serial steps for the balance of the stiffness matrix computations unique to the axisymmetric conical shell element are explicitly described in the NASTRAN Theoretical Manual section 5.9.5.7 (Summary of Procedures).

### 4.87.9.3 Mass Matrix Computation (Subroutine MCONE of Module SMA2)

$$[M_{ii}] = \begin{bmatrix} m_i & 0 & 0 & 0 & 0 & 0 \\ 0 & m_i & 0 & 0 & 0 & 0 \\ 0 & 0 & m_i & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad i = a \text{ or } b \quad (5)$$

where

$$\begin{aligned} m_a &= \pi \ell \left( \frac{r_b}{6} + \frac{r_a}{3} \right) (\rho t + \mu) \\ m_b &= \pi \ell \left( \frac{r_b}{3} + \frac{r_a}{6} \right) (\rho t + \mu) \end{aligned} \quad (6)$$

### 4.87.9.4 Element Load Calculations (Subroutine CONE of Module SSG1).

The Fourier coefficients of the temperatures are stored in the GPTT data block. The loads are generated by the elements, which reference the connected rings and harmonics indirectly by the grid point scalar indices. The scalar indices are used with the SIL (Scalar Index List) data block to obtain the temperatures. The following steps are used to generate the loads:

1. The data for a logical element are read from the EST data block. The harmonic



# STRUCTURAL ELEMENT DESCRIPTIONS

number,  $n$ , is extracted from the element ID,  $N_e$ , by the equation:

$$N_e = 1000N + (n + 1) , \quad (7)$$

where  $N$  is the total number of harmonics plus one in the problem.

The temperatures for this particular element and harmonic ( $T_n^a$  and  $T_n^b$ ) are extracted from the GPTT data block. (No default nonzero temperatures are allowed).

2. The following data are generated in the same manner as in the stiffness matrix routine, KCONE:

- $r_a, r_b, z_a, z_b$  - Ring locations
- $\ell$  - Linear distance
- $\sin\psi, \cos\psi$  - Inclination functions
- $[E]$  - Element-to-global transformation matrix (6x5)
- $t$  - Element thickness
- $[E_m]$  - Material matrix (3x3)
- $\alpha_s = \alpha_\phi = \alpha$  - Temperature coefficient

3. The geometry coefficients,  $I_{mn}$ , are calculated as in the stiffness matrix routine by the equation:

$$I_{mn} = \pi \int_0^\ell s^m r^{l-n} ds = \frac{\pi}{b} \int_{r_a}^{r_b} s^m r^{l-n} dr , \quad (8)$$

where  $r = a + bs$ ,  $a = r_a$ ,  $b = \frac{r_b - r_a}{\ell}$ .

4. The loads are generated in generalized coordinates,  $\{P_n^q\}$ , with the equations:

$$\{P_{1n}\} = n(I_{01}A_n + I_{11}B_n) , \quad (9)$$

# MODULE FUNCTIONAL DESCRIPTIONS

$$\{P_{2n}\} = n(I_{11}A_n + I_{21}B_n) , \quad (10)$$

$$\{P_{3n}\} = \sin\psi (I_{01}A_n + I_{11}B_n) , \quad (11)$$

$$\{P_{4n}\} = \sin\psi(I_{11}A_n + I_{21}B_n) + I_{00}C_n + I_{10}D_n , \quad (12)$$

$$\{P_{5n}\} = \cos\psi(I_{01}A_n + I_{11}B_n) , \quad (13)$$

$$\{P_{6n}\} = \cos\psi(I_{11}A_n + I_{21}B_n) , \quad (14)$$

$$\{P_{7n}\} = \cos\psi(I_{21}A_n + I_{31}B_n) , \quad (15)$$

$$\{P_{8n}\} = \cos\psi(I_{31}A_n + I_{41}B_n) , \quad (16)$$

$$\{P_{9n}\} = 0 , \quad (17)$$

$$\{P_{10,n}\} = 0 , \quad (18)$$

where

$$A_n = tE_{12}(\alpha_s T_n^a) + tE_{22}(\alpha_\phi T_n^a) , \quad (19)$$

$$B_n = \frac{t}{\ell} (T_n^b - T_n^a)(E_{12}\alpha_s + E_{22}\alpha_\phi) , \quad (20)$$

$$C_n = tE_{11}(\alpha_s T_n^a) + tE_{12}(\alpha_\phi T_n^a) , \quad (21)$$

$$D_n = \frac{t}{\ell} (T_n^b - T_n^a)(E_{11}\alpha_s + E_{12}\alpha_\phi) . \quad (22)$$

# STRUCTURAL ELEMENT DESCRIPTIONS

5. The transformation from generalized coordinates to element coordinates,  $[G_{qu}]$ , is calculated if no transverse shear material or thickness is given (i.e.,  $MAT ID_s = 0$  or  $t_s = 0$ ).

$$[H_{qu}] = [H]^{-1}, \quad (23)$$

where  $[H]$  is given explicitly in the stiffness matrix calculations.

6. If transverse shear exists, the shear matrix,  $[B]$ , is generated. Additional material terms are:

$[D] = I_b[E_b]$ , where  $[E_b]$  is the bending material  $3 \times 3$  matrix.

$G_{11} = G =$  Shear coefficient of transverse shear material.

The nonzero terms of  $[B]$  are:

$$B_{4,1} = B_{9,1} = \frac{1}{Q} [D_{12} n \cos \psi \left( \frac{1}{r_b} - \frac{1}{r_a} \right) - \frac{n}{2} \cos \psi \sin \psi \frac{I_{03}}{\pi} (D_{33} + 2D_{22})], \quad (24)$$

$$B_{4,2} = B_{9,2} = \frac{1}{Q} [D_{12} \frac{n \ell \cos \psi}{r_b} - \frac{1}{2} n \sin \psi \cos \psi \frac{I_{13}}{\pi} (3D_{33} + D_{22}) + \frac{3}{2} n D_{33} \cos \psi \frac{I_{02}}{\pi}], \quad (25)$$

$$B_{4,3} = B_{9,3} = \frac{1}{Q} \left[ \frac{1}{2} n^2 D_{33} \cos \psi \frac{I_{03}}{\pi} \right], \quad (26)$$

$$B_{4,4} = B_{9,4} = \frac{1}{Q} \left[ \frac{1}{2} n^2 D_{33} \cos \psi \frac{I_{13}}{\pi} \right], \quad (27)$$

$$B_{4,5} = B_{9,5} = \frac{1}{Q} [n^2 D_{12} \left( \frac{1}{r_b} - \frac{1}{r_a} \right) - n^2 \sin \psi \frac{I_{03}}{\pi} (2D_{33} + D_{22})], \quad (28)$$

$$B_{4,6} + B_{9,6} = \frac{1}{Q} [D_{12} \frac{n^2 \ell}{r_b} - n^2 \sin \psi \frac{I_{13}}{\pi} (2D_{33} + D_{22}) + \frac{I_{02}}{\pi} (2n^2 D_{33} + \sin^2 \psi D_{22})], \quad (29)$$

$$B_{4,7} = B_{9,7} = \frac{1}{Q} [2D_{11} (r_a - r_b) + D_{12} \frac{n^2 \ell^2}{r_b} + \frac{2I_{12}}{\pi} (2n^2 D_{33} + \sin^2 \psi D_{22}) - n^2 \sin \psi \frac{I_{23}}{\pi} (2D_{33} + D_{22})], \quad (30)$$



MODULE FUNCTIONAL DESCRIPTIONS

$$B_{4,8} = B_{9,8} = \frac{1}{Q} [-D_{11} 6\ell r_b + D_{12} \frac{n^2 \ell^3}{r_b} + \frac{3I_{22}}{\pi} (2n^2 D_{33} + \sin^2 \psi D_{22}) \quad (31)$$

$$- n^2 \sin \psi \frac{I_{33}}{\pi} (2D_{33} + D_{22})],$$

$$B_{4,9} = B_{9,9} = \frac{1}{Q} [-n \sin \psi \frac{I_{02}}{\pi} (D_{22} + D_{33})], \quad (32)$$

$$B_{4,10} = B_{9,10} = \frac{1}{Q} [n\ell(D_{12} + D_{33}) - n \sin \psi \frac{I_{12}}{\pi} (D_{22} + D_{33})], \quad (33)$$

where

$$Q = \ell t_s G_{11} \frac{r_a + r_b}{2} [1 + \frac{I_{02}}{\pi} \frac{n^2 D_{33} + \sin^2 \psi D_{22}}{\ell t_s G_{11} r_{ar}}] . \quad (34)$$

7. The transformation  $[\bar{H}]$  transforming displacements in the element coordinate system to displacements in generalized coordinates of the power series is:

$$[\bar{H}] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \frac{\cos \psi}{r_a} & 0 & 0 & 0 & \frac{n}{r_a} & 0 & 0 & 0 & 1 & 0 \\ 1 & \ell & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \ell & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \ell & \ell^2 & \ell^3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2\ell & 3\ell^3 & -1 & 0 \\ \frac{\cos \psi}{r_b} & \frac{\ell \cos \psi}{r_b} & 0 & 0 & \frac{n}{r_b} & \frac{n\ell}{r_b} & \frac{n\ell^2}{r_b} & \frac{n\ell^3}{r_b} & 1 & \ell \end{bmatrix} \quad (35)$$

# STRUCTURAL ELEMENT DESCRIPTIONS

8. The transformation  $[H_{qu}]$  for nonzero transverse shear is:

$$[H_{qu}] = ([\bar{H}] - [B])^{-1} \quad (10 \times 10) \quad . \quad (36)$$

9.  $[H_{qu}]$  is partitioned into two  $10 \times 5$  matrices

$$[H_{qu}] = [H_a \mid H_b] \quad . \quad (37)$$

10. The loads in global coordinates are calculated with:

$$\{P_a\} = [E][H_a]^T \{P_n^q\} \quad , \quad (38)$$

$$\{P_b\} = [E][H_b]^T \{P_n^q\} \quad . \quad (39)$$

## 4.87.9.5 Element Stress Calculations (Subroutines SCØNE1, SCØNE2, SCØNE3 of Module SDR2)

1. For each element the following quantities are calculated as in section 4.87.9.2:

$\ell$ ,  $\sin\psi$ ,  $\cos\psi$ ,  $[E]$ ,  $[H_a]$ ,  $[H_b]$ ,  $[H_{eq}]$ ,  $[H_{\psi q}]$ ,  $[H_{\chi q}]$  (using  $s = \frac{\ell}{2}$ ,  $r = \frac{r_a + r_b}{2}$ ).

2. Using the material properties, the following matrices are calculated:

$$[E_m] = \frac{E_1}{(1 - \nu_1^2)} \begin{bmatrix} 1 & -\nu_1 & 0 \\ -\nu_1 & 1 & 0 \\ 0 & 0 & \frac{1 - \nu_1}{2} \end{bmatrix} \quad , \quad (40)$$

# MODULE FUNCTIONAL DESCRIPTIONS

$$[D_b] = \frac{E_2 I}{(1 - \nu_2^2)} \begin{bmatrix} 1 & -\nu_2 & 0 \\ -\nu_2 & 1 & 0 \\ 0 & 0 & \frac{1 - \nu_2}{2} \end{bmatrix}, \quad (41)$$

$$[G_s] = \begin{bmatrix} G_3 & 0 \\ 0 & G_3 \end{bmatrix}, \quad (42)$$

where  $[E_m]$ ,  $[D_b]$  and  $[G_s]$  are computed for membrane, bending and shear materials respectively.

3. The stress matrices are then calculated:

$$[K_s] = \begin{bmatrix} [E_m] & [H_{eq}] \\ - & - & - & - \\ [D] & [H_{xq}] \\ - & - & - & - \\ t_s [G_s] & [H_{\alpha q}] \end{bmatrix}, \quad (8 \times 10) \quad (43)$$

$$[S_a] = [K_s][H_a][E], (8 \times 6) \quad (44)$$

$$[S_b] = [K_s][H_b][E], (8 \times 6). \quad (45)$$

$$St_1 = \alpha_1 E_{11} + \alpha_2 E_{12}$$

$$St_2 = \alpha_1 E_{12} + \alpha_2 E_{12}$$

$$(\alpha_1 = \alpha_2 \text{ for type 1 materials})$$

4. Each entry in the EST data block contains data pertaining to the  $n^{\text{th}}$  harmonic motion of the CØNEAX element. The elements in the EST are ordered by harmonic and CØNEAX I.D. number. All harmonic elements for each CØNEAX are grouped together.

a. When the harmonic,  $n$ , of an element is zero, this indicates it is the first of a group of elements. Storage space is allotted for fourteen 8 by 1 vectors defining the element forces at points. Two UGV vector data blocks must be used to calculate stresses on points. These data blocks correspond to the two subcases "C" and "S" and are solved simultaneously using the same data.



# STRUCTURAL ELEMENT DESCRIPTIONS

b. Using the  $[S_a]$  and  $[S_b]$  matrices and the 6 by 1 displacement vectors,  $\{u\}$ , by their SIL numbers, stress and force vectors are computed.

If  $n \neq 0$ :

$$\begin{Bmatrix} \bar{\sigma}_{sn}^c \\ \bar{\sigma}_{\phi n}^c \\ \bar{\sigma}_{s\phi n}^c \\ M_{sn}^c \\ M_{\phi n}^c \\ M_{s\phi n}^c \\ V_{sn}^c \\ V_{\phi n}^c \end{Bmatrix} = [S_{an}] \{u_{an}^c\} + [S_{bn}] \{u_{bn}^c\}, \quad (46)$$

$$\begin{Bmatrix} \bar{\sigma}_{sn}^s \\ \bar{\sigma}_{\phi n}^s \\ \bar{\sigma}_{s\phi n}^s \\ M_{sn}^s \\ M_{\phi n}^s \\ M_{s\phi n}^s \\ V_{sn}^s \\ V_{\phi n}^s \end{Bmatrix} = [S_{an}] \{u_{an}^s\} + [S_{bn}] \{u_{bn}^s\}. \quad (47)$$

# MODULE FUNCTIONAL DESCRIPTIONS

If  $n = 0$ :

$$\begin{Bmatrix} \bar{\sigma}_{so} \\ \bar{\sigma}_{\phi 0} \\ \bar{\sigma}_{s\phi 0} \\ M_{so} \\ M_{\phi 0} \\ M_{s\phi 0} \\ V_{so} \\ V_{\phi 0} \end{Bmatrix} = [S_{ao}] (\{u_{ao}^s\} + \{u_{ao}^c\}) - [S_{bo}] (\{u_{bo}^s\} + \{u_{bo}^c\}). \quad (48)$$

c. The temperature effects are added using the GPTT data for the two subcases,  $(T_{\alpha}^c \text{ and } T_{\alpha}^s)$

$$(\sigma_{sn}^{c'}) = \sigma_{sn}^c + S_{t1} \bar{T} \quad (49)$$

$$(\sigma_n^{c'}) = \sigma_{\phi n}^c + S_{t2} \bar{T} \quad (50)$$

$$(\sigma_{s\phi n}^{c'}) = \sigma_{s\phi n}^c \quad (51)$$

where

$$\bar{T} = \frac{1}{2} (T_a^n + T_b^n) - T_o, \quad n = 0$$

$$\bar{T} = \frac{1}{2} (T_a^n + T_b^n) \quad n \neq 0$$

The same equations are used for the "S" set and when  $n = 0$ .

# MODULE FUNCTIONAL DESCRIPTIONS

d. The harmonic stresses are calculated by the equations  
for  $i = 1, 2, 3$ :

$$\sigma_{sni}^C = (\sigma_{sn}^{C'}) + \frac{M_{sn}^C c_i}{I} \quad , \quad (52)$$

$$\sigma_{\phi n}^C = (\sigma_{\phi n}^{C'}) + \frac{M_{\phi n}^C c_i}{I} \quad , \quad (53)$$

$$\sigma_{s\phi n}^C = (\sigma_{s\phi n}^{C'}) + \frac{M_{s\phi n}^C c_i}{I} \quad . \quad (54)$$



# STRUCTURAL ELEMENT DESCRIPTIONS

The equations are repeated for the S set and when  $n = 0$ .

e. Principal stresses ( $\sigma_1, \sigma_2, \theta, \tau_{\max}$  etc.) are calculated as with the TRIAL or QUAD1 element, except that when  $n \neq 0$  the data are calculated for both the S and C sets.

f. The incremental element stresses or forces for the points on the cone are calculated from the following equations for  $j = 1, 2, \dots, 14$ :

For  $n \neq 0$ :

$$\delta\sigma_{sj} = (\sigma_{sn}^C)' \cos(n\phi_j) + (\sigma_{sn}^S)' \sin(n\phi_j), \quad (55)$$

$$\delta\sigma_{\phi j} = (\sigma_{\phi n}^C)' \cos(n\phi_j) + (\sigma_{\phi n}^S)' \sin(n\phi_j), \quad (56)$$

$$\delta\sigma_{s\phi j} = (\sigma_{s\phi n}^C)' \sin(n\phi_j) - (\sigma_{s\phi n}^S)' \cos(n\phi_j), \quad (57)$$

$$\delta M_{sj} = M_{sn}^C \cos(n\phi_j) + M_{sn}^S \sin(n\phi_j), \quad (58)$$

$$\delta M_{\phi j} = M_{\phi n}^C \cos(n\phi_j) + M_{\phi n}^S \sin(n\phi_j), \quad (59)$$

$$\delta M_{s\phi j} = M_{s\phi n}^C \sin(n\phi_j) - M_{s\phi n}^S \cos(n\phi_j), \quad (60)$$

$$\delta V_{sj} = V_{sn}^C \cos(n\phi_j) + V_{sn}^S \sin(n\phi_j), \quad (61)$$

$$\delta V_{\phi j} = V_{\phi n}^C \sin(n\phi_j) - V_{\phi n}^S \cos(n\phi_j). \quad (62)$$

g. The incremental stress and force values are added to the running sums for the points. After the last element is calculated ( $n = N$ ), the forces and stresses for the points are calculated and output. The equations are identical to steps d and e of this section except that 1) the "S" and "C" sets are not used and 2) up to 14 points may be calculated for output for each physical element.

Since the user may leave some spaces blank on the property card for this element, only one of the  $\phi_j = 0$  points is used in the calculation.

## STRUCTURAL ELEMENT DESCRIPTIONS

### 4.87.9.6 Differential Stiffness Matrix Calculations (Subroutine DCØNE of Module DSMG1)

The data input from the ECPT to the DCØNE subroutine are the same as those given in section 4.87.9.1. Additional data for the generation of the differential stiffness matrix are as follows:

$\{u_a^0\}, \{u_b^0\}$  - Displacement vectors of the zero harmonic (extracted from the UGV data block)

$T_a^0, T_b^0$  - Element loading temperatures of the zero harmonic (extracted from the GPTT data block)

The first part of the calculations involves calculation of the element force components. The following steps are performed with harmonic number  $n = 0$ .

1. The 10 by 10 transformation matrix  $[H_{uq}^0]$  is computed from:

$$[H_{uq}^n] = [\bar{H}_{uq}] + \{H_{uY}\} \{H_{Ysq}\}^T \quad (63)$$

where  $[\bar{H}_{uq}]$ , a 10 by 10 matrix, and  $\{H_{uY}\}$ , a column vector, are derived in the NASTRAN Theoretical Manual, section 5.9.5.3, and  $\{H_{Ysq}\}^T$ , a row vector, is explicitly written out in Equation 85 of section 5.9 of the NASTRAN Theoretical Manual.

2. The 10 generalized displacement quantities  $q_i$  are

$$\{q\} = [H_{uq}^0]^{-1} \begin{pmatrix} [E]^T \{u_a^0\} \\ [E]^T \{u_b^0\} \end{pmatrix} \quad (64)$$

where  $\{u_a^0\}$  and  $\{u_b^0\}$  are the 6 by 1 displacement vectors, and  $[E]$  is calculated as in Equation 4 of section 4.87.9.2.

3. The strain coefficients are

$$\begin{pmatrix} \Delta\epsilon_s \\ \Delta\epsilon_\phi \\ \Delta\epsilon_{s\phi} \end{pmatrix} = \{\alpha\} \frac{(T_b^c - T_a^0)}{\ell} \quad (65)$$

# STRUCTURAL ELEMENT DESCRIPTIONS

$$\begin{pmatrix} \epsilon_s \\ \epsilon_\phi \\ \epsilon_{s\phi} \end{pmatrix} = T_a^0 \{\alpha\} \quad (66)$$

where  $T_a^0$  and  $T_b^0$  are the loading temperatures at the grid points, the  $\{\alpha\}$  vector is obtained from the MPT data block via subroutine MAT and  $\ell$  is calculated as in Equation 1 of section 4.87.9.2.

4. The force coefficients are calculated:

$$a_0 = t_m G_{12} (\sin \psi q_3 + \cos \psi q_5) , \quad (67)$$

$$a_1 = t_m G_{12} (\sin \psi q_4 + \cos \psi q_6) , \quad (68)$$

$$a_2 = t_m G_{12} \cos \psi q_7 , \quad (69)$$

$$a_3 = t_m G_{12} \cos \psi q_8 , \quad (70)$$

$$b_0 = t_m G_{22} (\sin \psi q_3 + \cos \psi q_5) , \quad (71)$$

$$b_1 = t_m G_{22} (\sin \psi q_4 + \cos \psi q_6) , \quad (72)$$

$$b_2 = t_m G_{22} \cos \psi q_7 , \quad (73)$$

$$b_3 = t_m G_{22} \cos \psi q_8 , \quad (74)$$

$$c_0 = t_m G_{11} (q_4 - \epsilon_s) - t_m G_{12} \epsilon_\phi , \quad (75)$$

$$c_1 = -t_m G_{11} \Delta \epsilon_s - t_m G_{12} \Delta \epsilon_\phi , \quad (76)$$

$$d_0 = t_m G_{12} (q_4 - \epsilon_s) - t_m G_{22} \epsilon_\phi , \quad (77)$$

$$d_1 = -t_m G_{12} \Delta \epsilon_s - t_m G_{22} \Delta \epsilon_\phi , \quad (78)$$



# STRUCTURAL ELEMENT DESCRIPTIONS

where  $G_{11}$ ,  $G_{12}$  and  $G_{22}$  are elements of the 3 by 3 symmetric material properties matrix,  $[G]$ .

5. The geometry coefficients are calculated:

$$I_{mn} = \int_0^{\ell} s^m r^{1-n} ds \quad \begin{cases} m = 0, 1, \dots, 9 \\ n = 0, 1, 2, 3 \end{cases} ; \quad (79)$$

where

$$r = r_a + \left( \frac{r_b - r_a}{\ell} \right) s . \quad (80)$$

An explicit formula for the evaluation of  $I_{mn}$  is given in the NASTRAN Theoretical Manual, section 5.9.5.8.

6. The following coefficients for the computation of the differential stiffness matrix are calculated:

$$\begin{aligned} A_{mn} = & a_0 I_{m, n+1} + a_1 I_{m+1, n+1} + a_2 I_{m+2, n+1} \\ & + a_3 I_{m+3, n+1} + c_0 I_{m, n} + c_1 I_{m+1, n} , \end{aligned} \quad (81)$$

$$\begin{aligned} B_{mn} = & b_0 I_{m, n+1} + b_1 I_{m+1, n+1} + b_2 I_{m+2, n+1} \\ & + b_3 I_{m+3, n+1} + d_0 I_{m, n} + d_1 I_{m+1, n} , \end{aligned} \quad (82)$$

$$C_{mn} = A_{mn} + B_{mn} , \quad (83)$$

where  $m = 0, 1, \dots, 6$ ;  $n = 0, 1, 2$ .

Note: The index  $n$  used above is a dummy index and is not the harmonic number.

The second part of the calculations involves generating the differential stiffness matrix.

The remaining steps use  $n$  as the harmonic number of the element.

1. The nonzero elements of the symmetric differential stiffness matrix  $[k^{qd}]$ , in generalized coordinates, are given in Table 1c below.

# STRUCTURAL ELEMENT DESCRIPTIONS

Table 1c. Nonzero Elements of the Differential Stiffness Matrix,  $[K^{qd}]$ .

$$K_{11}^{qd} = \cos^2 \psi_{B_{02}} + \frac{1}{4} \sin^2 \psi_{C_{02}}$$

$$K_{12}^{qd} = \cos^2 \psi_{B_{12}} + \frac{1}{4} \sin \psi_{C_{01}} + \frac{1}{4} \sin^2 \psi_{C_{12}}$$

$$K_{13}^{qd} = \frac{1}{4} n \sin \psi_{C_{02}}$$

$$K_{14}^{qd} = \frac{1}{4} n \sin \psi_{C_{12}}$$

$$K_{15}^{qd} = n \cos \psi_{B_{02}}$$

$$K_{16}^{qd} = n \cos \psi_{B_{12}}$$

$$K_{17}^{qd} = n \cos \psi_{B_{22}}$$

$$K_{18}^{qd} = n \cos \psi_{B_{32}}$$

---


$$K_{22}^{qd} = \cos^2 \psi_{B_{22}} + \frac{1}{4} C_{00} + \frac{1}{2} \sin \psi_{C_{11}} + \frac{1}{4} \sin^2 \psi_{C_{22}}$$

$$K_{23}^{qd} = \frac{1}{4} n C_{01} + \frac{1}{4} n \sin \psi_{C_{12}}$$

$$K_{24}^{qd} = \frac{1}{4} n C_{11} + \frac{1}{4} n \sin \psi_{C_{22}}$$

$$K_{25}^{qd} = n \cos \psi_{B_{12}}$$

$$K_{26}^{qd} = n \cos \psi_{B_{22}}$$

$$K_{27}^{qd} = n \cos \psi_{B_{32}}$$

$$K_{28}^{qd} = n \cos \psi_{B_{42}}$$

---


$$K_{33}^{qd} = \frac{1}{4} n^2 C_{02}$$

$$K_{34}^{qd} = \frac{1}{4} n^2 C_{12}$$


---

# STRUCTURAL ELEMENT DESCRIPTIONS

Table 1c (con'd). Elements of the Differential Stiffness Matrix,  $[K^{qd}]$ .

$$K_{44}^{qd} = \frac{1}{4} n^2 C_{22}$$

---


$$K_{55}^{qd} = n^2 B_{02}$$

$$K_{56}^{qd} = n^2 B_{12}$$

$$K_{57}^{qd} = n^2 B_{22}$$

$$K_{58}^{qd} = n^2 B_{32}$$

---


$$K_{66}^{qd} = A_{00} + n^2 B_{22}$$

$$K_{67}^{qd} = 2A_{10} + n^2 B_{32}$$

$$K_{68}^{qd} = 3A_{20} + n^2 B_{42}$$

---


$$K_{77}^{qd} = 4A_{20} + n^2 B_{42}$$

$$K_{78}^{qd} = 6A_{30} + n^2 B_{52}$$

---


$$K_{88}^{qd} = 9A_{40} + n^2 B_{62}$$


---

The formulas for  $n = 0$  are the same as in Table 1c except that they are all multiplied by 2. The nonzero terms for  $n = 0$  fall into two uncoupled sets which are

(11)    (12)  
         (22)

~~~~~  
Effect on  
Twisting

(66)    (67)    (68)  
         (77)    (78)  
                 (88)

~~~~~  
Effect on Axisymmetric Deformation

2. The 10 by 10 transformation matrix,  $[H_{uq}^n]$ , from generalized coordinates to element coordinates, for the  $n^{\text{th}}$  harmonic is computed as in Equation 63. The matrix is inverted and partitioned into two 10 by 5 matrices:



# STRUCTURAL ELEMENT DESCRIPTIONS

$$[H_{uq}^n]^{-1} \Rightarrow [H_a \mid H_b] \quad . \quad (84)$$

3. The 6 by 6 differential stiffness matrices in global coordinates are:

$$[K_{ij}^d] = [E] [H_i]^T [K^{qd}] [H_j] [E]^T \quad , \quad (85)$$

where  $i$  = pivot grid point;  $j$  = a, b; and  $[E]$  is computed as in Equation 4 of section 4.87.9.2.

## MODULE FUNCTIONAL DESCRIPTIONS

### 4.87.10 The TRIARG Element

#### 4.87.10.1 Input Data for the TRIARG Element

1. The ECPT/EST entries for the axisymmetric triangular ring (TRIARG) element are:

| <u>Symbol</u>   | <u>Descriptions</u>  |
|---|--|
| $SIL_1, SIL_2, SIL_3$   | Scalar index numbers for the three grid points.  |
| $\gamma$  | Material property orientation angle (degrees)  |
| Mat I.D.  | Material property identification number.   |
| $\left. \begin{array}{l} N_1, X_1, Y_1, Z_1 \\ N_2, X_2, Y_2, Z_2 \\ N_3, X_3, Y_3, Z_3 \end{array} \right\}$ | Local coordinate system number and location in basic coordinates of the three grid points. |
| $t_\mu$   | Element temperature for material properties.   |

For this element,  $Y_i$  must equal zero for  $i = 1, 2$  and  $3$  and, we define:

$$\{R_s\} = \begin{Bmatrix} X_1 \\ X_2 \\ X_3 \end{Bmatrix}, \quad (1)$$

$$\{Z_s\} = \begin{Bmatrix} Z_{s1} \\ Z_{s2} \\ Z_{s3} \end{Bmatrix} = \begin{Bmatrix} Z_1 \\ Z_2 \\ Z_3 \end{Bmatrix}. \quad (2)$$

#### 2. Coordinate system data

The location  $(X_i, Y_i, Z_i)$  and local coordinate system number  $(N_i)$  of each grid point are used to calculate the 3 by 3 global-to-basic coordinate system transformation matrices,  $[T_i]$ ,  $i = 1, 2, 3$ .

#### 3. Material data

The material property identification number, Mat I.D., and the element temperature for material properties,  $t_\mu$ , are used to select the following data items. For this element, material properties may be defined on a MAT1 or MAT3 but not a MAT2 bulk data card.

# STRUCTURAL ELEMENT DESCRIPTIONS

| <u>Symbol</u>                             | <u>Description</u>  |
|---|---|
| $E_r, E_\theta, E_z$                      | Young's moduli in the radial, tangential and axial directions respectively. |
| $\nu_{r\theta}, \nu_{\theta z}, \nu_{zr}$ | Poisson's ratios in the three directions indicated.                         |
| $\rho$                                    | Mass density  |
| $G_{r\theta}, G_{\theta z}, G_{rz}$       | Shear moduli in the three directions indicated.                             |
| $\alpha_r, \alpha_\theta, \alpha_z$       | Coefficients of thermal expansion in the three directions indicated.        |
| $T_0$                                     | Thermal expansion reference temperature.                                    |
| $g_e$                                     | Structural element damping coefficient.                                     |

## 4.87.10.2 General Geometric Calculations

1. Local coordinate calculations are:

$$Z_{\min} = \text{minimum of } (Z_{s1}, Z_{s2}, Z_{s3}), \quad (3)$$

$$\{R_L\} = \{R_S\}, \quad (4)$$

$$\{Z_L\} = \{Z_S\} - \begin{pmatrix} Z_{\min} \\ Z_{\min} \\ Z_{\min} \end{pmatrix}. \quad (5)$$

2. The transformation from field coordinates to grid point degrees of freedom is given by ( $R_{Li}$  and  $Z_{Li}$  are the  $i^{\text{th}}$  components of  $\{R_L\}$  and  $\{Z_L\}$  respectively):

$$[\bar{T}_{\beta q}] = \begin{bmatrix} 1 & R_{L1} & Z_{L1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & R_{L1} & Z_{L1} \\ 1 & R_{L2} & Z_{L2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & R_{L2} & Z_{L2} \\ 1 & R_{L3} & Z_{L3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & R_{L3} & Z_{L3} \end{bmatrix}, \quad (6)$$

6x6



# MODULE FUNCTIONAL DESCRIPTIONS

$$[\Gamma_{\beta q}] = [\bar{\Gamma}_{\beta q}]^{-1}. \quad (7)$$

3. The transformation matrix from two to three degrees of freedom per point is:

$$[\Gamma_{qs}] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad 6 \times 9. \quad (8)$$

## 4.87.10.3 Integral Calculations

1. The integrals over the area of the cross-section are of the form:

$$\delta_{ij} = r^i z^j dz dr, \quad (9)$$

for the values:

$$\delta_{-10}, \delta_{-11}, \delta_{00}, \delta_{10}, \delta_{20}, \delta_{30}, \delta_{01}, \delta_{11}, \delta_{21}, \delta_{02}, \delta_{12}, \delta_{-12}.$$

To accomplish this we integrate in two parts:

a. From line  $Z = K_{12}r + m_{12}$  to line  $Z = K_{13}r + m_{13}$

where

$$K_{ij} = \frac{Z_{Lj} - Z_{Li}}{R_{Lj} - R_{Li}}, \quad (10)$$

and

$$m_{ij} = -\frac{R_{Li}Z_{Lj} - R_{Lj}Z_{Li}}{R_{Lj} - R_{Li}}, \quad (11)$$

# STRUCTURAL ELEMENT DESCRIPTIONS

and from point  $R_{L1}$  to point  $R_{L3}$ .

b. From line  $Z = K_{12}r + m_{12}$  to  $Z = K_{32}r + m_{32}$  and from point  $R_{L3}$  to  $R_{L2}$ .

For the case where

$$R_{L1} = R_{L2} \text{ or } \left| \frac{R_{L2} - R_{L1}}{R_{L2}} \right| < 10^{-5}$$

we must integrate differently, that is, from line  $Z = K_{32}r + m_{32}$  to  $Z = K_{13}r + m_{13}$  and from  $R_{L1}$  to  $R_{L3}$ .

2. After the integrals are computed, a check is made to determine if an excessive amount of round-off error occurred. If round-off was excessive, an approximate integral can be calculated.

These tests are:

If any  $\delta_{ij} < 0$ , then approximation must be used.

If  $\sigma_{12} \leq \sigma_{02}$ , or  $\delta_{-12} \geq \delta_{12}$ , or  $\delta_{-12} > \delta_{02}$ , then approximation must be used.

If  $\Delta r \leq \hat{r}$  or  $\Delta Z \leq \hat{Z}$ , then approximation must be used. The terms  $\Delta r$ ,  $\Delta Z$ ,  $\hat{r}$  and  $\hat{Z}$  are:

$$\Delta r = \max.(|R_{L1} - R_{L2}|, |R_{L2} - R_{L3}|, |R_{L3} - R_{L1}|), \quad (12)$$

$$\hat{r} = [\min.(R_{L1}, R_{L2}, R_{L3})]/10, \quad (13)$$

$$\Delta Z = \max.(|Z_{L1} - Z_{L2}|, |Z_{L2} - Z_{L3}|, |Z_{L3} - Z_{L1}|), \quad (14)$$

$$\hat{Z} = [\min.(Z_{L1}, Z_{L2}, Z_{L3})]/10. \quad (15)$$

# MODULE FUNCTIONAL DESCRIPTIONS

The approximation is:

$$\sigma_{ij} = (r_a)^i (z_a)^j A, \quad (16)$$

where

$$r_a = \frac{1}{3} [R_{L1} + R_{L2} + R_{L3}], \quad (17)$$

$$z_a = \frac{1}{3} [Z_{L1} + Z_{L2} + Z_{L3}], \quad (18)$$

$$A = \frac{1}{2} [R_{L1} (Z_{L2} - Z_{L3}) + R_{L2} (Z_{L3} - Z_{L1}) + R_{L3} (Z_{L1} - Z_{L2})]. \quad (19)$$

3. Form the matrix of integrals:

$$[\tilde{D}] = 2\pi \begin{bmatrix} 0 & \delta_{10} & 0 & 0 & 0 & 0 \\ \delta_{00} & \delta_{10} & \delta_{01} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \delta_{10} \\ 0 & 0 & \delta_{10} & 0 & \delta_{10} & 0 \end{bmatrix}_{4 \times 6} \quad (20)$$

## 4.87.10.4 Elastic Constants Matrix Calculations

1. Generate the transformation from material axis to element geometric axis:

$$[T_{eo}] = \begin{bmatrix} \cos^2 \gamma & 0 & \sin^2 \gamma & \sin \gamma \cos \gamma \\ 0 & 1 & 0 & 0 \\ \sin^2 \gamma & 0 & \cos^2 \gamma & -\sin \gamma \cos \gamma \\ -2 \sin \gamma \cos \gamma & 0 & 2 \sin \gamma \cos \gamma & \cos^2 \gamma - \sin^2 \gamma \end{bmatrix}_{4 \times 4} \quad (21)$$

2. Generate the matrix of elastic constants for an orthotropic body with respect to cylindrical coordinates:



# STRUCTURAL ELEMENT DESCRIPTIONS

$$[E_m] = \frac{1}{\Delta} \begin{bmatrix} E_r(1-\nu_{\theta z}\nu_{z\theta}) & E_r(\nu_{\theta r}+\nu_{zr}\nu_{\theta z}) & E_r(\nu_{zr}+\nu_{\theta r}\nu_{z\theta}) & 0 \\ E_{\theta}(1-\nu_{rz}\nu_{zr}) & E_{\theta}(\nu_{z\theta}+\nu_{r\theta}\nu_{zr}) & 0 & 0 \\ E_z(1-\nu_{r\theta}\nu_{\theta r}) & 0 & 0 & G_{rz}\Delta \end{bmatrix} \quad (22)$$

where

$$\nu_{\theta r} = \nu_{r\theta} E_{\theta}/E_r \quad (23)$$

$$\nu_{z\theta} = \nu_{\theta z} E_z/E_{\theta} \quad (24)$$

$$\nu_{rz} = \nu_{zr} E_r/E_z \quad (25)$$

$$\Delta = 1 - \nu_{r\theta}\nu_{\theta r} - \nu_{\theta z}\nu_{z\theta} - \nu_{zr}\nu_{rz} - \nu_{r\theta}\nu_{\theta z}\nu_{zr} - \nu_{rz}\nu_{\theta r}\nu_{z\theta} \quad (26)$$

3. Calculate the elastic constants matrix in element geometric axes:

$$[E_g] = [T_{eo}]^T [E_m] [T_{eo}] \quad (27)$$

## 4.87.10.5 Stiffness Matrix Generation (Subroutine KTRIRG of Module SMA1)

1. Generate the element stiffness matrix in field coordinates:

# MODULE FUNCTIONAL DESCRIPTIONS

$$[\tilde{K}] =$$

$$2\pi \begin{bmatrix} E_{22}\delta_{-10} & & & & & \\ (E_{12}+E_{22})\delta_{00} & (E_{11}+2E_{12}+E_{22})\delta_{10} & & & & \\ E_{22}\delta_{-11}+E_{24}\delta_{00} & (E_{12}+E_{22})\delta_{01}+(E_{14}+E_{24})\delta_{10} & E_{22}\delta_{-12}+2E_{24}\delta_{01}+E_{44}\delta_{10} & & & \\ 0 & 0 & 0 & 0 & & \\ E_{24}\delta_{00} & (E_{14}+E_{24})\delta_{10} & E_{24}\delta_{01}+E_{44}\delta_{10} & 0 & E_{44}\delta_{10} & \\ E_{23}\delta_{00} & (E_{13}+E_{23})\delta_{10} & E_{23}\delta_{01}+E_{34}\delta_{10} & 0 & E_{34}\delta_{10} & E_{33}\delta_{10} \end{bmatrix}_{6 \times 6} \quad (28)$$

where  $E_{ij}$  is an element of  $[E_g]$ .

2. Transform the element stiffness matrix from field coordinates to grid point degrees of freedom:

$$[\bar{K}] = [\Gamma_{\beta q}]^T [\tilde{K}] [\Gamma_{\beta q}]. \quad (29)$$

3. Transform the element stiffness matrix from two to three degrees of freedom per point.

$$[K] = [\Gamma_{qs}]^T [\bar{K}] [\Gamma_{qs}]. \quad (30)$$

4. The 3 by 3 partitions  $[K_{pj}^3]$  of  $[K]$  corresponding to the pivot point  $p$  are transformed:

$$[K_{pj}^3] = [T_p]^T [K_{pj}^3] [T_j], \quad j = 1, 2, 3 \quad (31)$$

5. Finally these 3 by 3 partitions are expanded to 6 by 6 partitions:

# STRUCTURAL ELEMENT DESCRIPTIONS

$$[K_{pj}] = \begin{bmatrix} k_{pj}^3 & \vdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \vdots & 0 \end{bmatrix} \quad (32)$$

## 4.87.10.6 Mass Matrix Calculations (Subroutine MTRIRG of Module SMA2)

1. Generate the consistent mass matrix in field coordinates:

$$[\tilde{M}] = 2\pi \begin{bmatrix} \bar{m}_{11}^{\delta_{10}} & & & & & \\ \bar{m}_{11}^{\delta_{20}} & \bar{m}_{11}^{\delta_{30}} & & & & \\ \bar{m}_{11}^{\delta_{11}} & \bar{m}_{11}^{\delta_{21}} & \bar{m}_{11}^{\delta_{12}} & & & \\ 0 & 0 & 0 & \bar{m}_{22}^{\delta_{10}} & & \\ 0 & 0 & 0 & \bar{m}_{22}^{\delta_{20}} & \bar{m}_{22}^{\delta_{30}} & \\ 0 & 0 & 0 & \bar{m}_{22}^{\delta_{11}} & \bar{m}_{22}^{\delta_{21}} & \bar{m}_{22}^{\delta_{12}} \end{bmatrix} \quad \begin{matrix} \text{(Symmetric)} \\ \\ \\ \end{matrix} \quad (33)$$

6x6

where

$$[\bar{m}] = \begin{bmatrix} \rho & 0 \\ 0 & \rho \end{bmatrix} = \begin{bmatrix} \bar{m}_{11} & 0 \\ 0 & \bar{m}_{22} \end{bmatrix}. \quad (34)$$

2. Transform the mass matrix from field coordinates to grid point degrees of freedom:

$$[\bar{M}] = [r_{\beta q}]^T [\tilde{M}] [r_{\beta q}]. \quad (35)$$

3. Transform the mass matrix from two to three degrees of freedom per point:

$$[M] = [r_{qs}]^T [\bar{M}] [r_{qs}]. \quad (36)$$



## MODULE FUNCTIONAL DESCRIPTIONS

4. The 6 by 6 partitions,  $[M_{pj}]$ , are calculated as in Equations 31 and 32.

### 4.87.10.7 Thermal Load Calculations (Subroutine TTRIRG of Module SSG1)

1. Form the vector of thermal strains:

$$\{\tilde{\alpha}\} = (T_{avg} - T_o) \begin{pmatrix} \alpha_r \\ \alpha_\theta \\ \alpha_z \\ 0 \end{pmatrix} \quad (4 \times 1). \quad (37)$$

where  $T_{avg}$  is the average loading temperature at the grid points.

2. Compute thermal load vector in grid point degrees of freedom:

$$\{\bar{F}_T\} = [r_{\beta q}]^T [\tilde{D}]^T [E_g] \{\alpha\} \quad (6 \times 1). \quad (38)$$

3. Transform thermal load from two to three degrees of freedom per point

$$\{F_T\} = [r_{qs}]^T \{\bar{F}_T\} \quad (9 \times 1). \quad (39)$$

4. Each partition,  $\{F_T^3\}$ , of length 3 of  $\{F_T\}$  is transformed to global coordinates by

$$\{F_T^3\}_g = [T_i]^T \{F_T^3\}. \quad (40)$$

5. These vectors are added to the overall load vector,  $\{P_g\}$ .

### 4.87.10.8 Element Force and Stress Calculations (Subroutines STRIR1 and STRIR2 of Module SDR2)

Element stress and force data items are calculated in two phases. The first phase (subroutine STRIR1) calculates the element stiffness and stress matrices. The second phase (subroutine STRIR2) calculates the actual element forces and stresses from the various subcase displacement vectors.

**"Page missing from available version"**

# MODULE FUNCTIONAL DESCRIPTIONS

## 4.87.10.9 Thermal Analysis Calculations for the TRIARG and TRAPRG Elements

If a heat transfer problem is being solved, special code and subroutines are used for these elements. The following checks are made on the geometry:

$$\left. \begin{array}{l} y = 0 \\ x_i > 0 \end{array} \right\} i = 1, 2, 3 \text{ (or 4)}.$$

If these conditions are not met, a fatal error exists. The order of the grid points is also checked. The following equation must be true:

$$(x_s - x_r)(z_t - z_s) - (x_t - x_s)(z_s - z_r) > 0,$$

where the indices r, s, t correspond to three grid points in the element where

r, s, t = 1, 2, 3 for triangles,

$$\text{or } \left. \begin{array}{l} s, t = 1, 2, 3 \\ 2, 3, 4 \\ 3, 4, 1 \\ 4, 1, 2 \end{array} \right\} \text{ for trapezoids.}$$

The conduction matrix for TRIARG is computed by calling for a TRMEM, whose thickness is calculated from

$$t = 2\pi(x_1 + x_2 + x_3)/3.$$

To compute the conduction matrix for TRAPRG, it will be divided into overlapping triangles. The mapping is exactly the same as the mapping of quadrilaterals into triangles. The material orientation angles for the triangles must be computed as was done for QDMEM. The thickness of each of the triangles is given by

$$t = \pi(x_r + x_s + x_t)/3,$$

where r, s, t, are the three vertex indices used from the mapping matrix. The KTRMEM subroutine is then called four times.



# STRUCTURAL ELEMENT DESCRIPTIONS

$$\left. \begin{array}{l} y = 0 \\ x_i > 0 \end{array} \right\} i = 1, 2, 3 \text{ (or 4).}$$

If these conditions are not met, a fatal error exists. The order of the grid points is also checked. The following equation must be true:

$$(x_s - x_r)(z_t - z_s) - (x_t - x_s)(z_s - z_r) > 0,$$

where the indices r, s, t correspond to three grid points in the element where

r, s, t = 1, 2, 3 for triangles,

$$\left. \begin{array}{l} \text{or } r, s, t = 1, 2, 3 \\ 2, 3, 4 \\ 3, 4, 1 \\ 4, 1, 2 \end{array} \right\} \text{ for trapezoids.}$$

The thermal "stiffness" matrices are generated by subroutine HRING of module SMA1. The conduction matrix for TRIARG is computed by calling for a TRMEM, whose thickness is calculated from

$$t = 2\pi(x_1 + x_2 + x_3)/3.$$

To compute the conduction matrix for TRAPRG, it will be divided into overlapping triangles. The mapping is exactly the same as the mapping of quadrilaterals into triangles. The material orientation angles for the triangles must be computed as was done for QDMEM. The thickness of each of the triangles is given by:

$$t = \pi(x_r + x_s + x_t)/3,$$

where r, s, t are the three vertex indices used from the mapping matrix. The KTRMEM subroutine is then called four times.

The thermal "mass" matrices are generated in a similar manner. Subroutine MRING in module SMA2 rearranges the ECPT data into the TRMEM format with the equivalent "thickness" given by:

$$t = \frac{2\pi}{3} (x_1 + x_2 + x_3).$$

## MODULE FUNCTIONAL DESCRIPTIONS

Subroutine MASSTQ generates the scalar mass terms for each triangle and inserts them into the BGG matrix. The TRIAG element is converted into a triangle once. The TRAPRG element is arranged into the TRMEM format for each of its four overlapping subtriangles; each triangle having a different equivalent "thickness,"  $t$ , given by the equation above.

The thermal "stress" output data calculations are performed by subroutines SDHTF1, SDHTF2, and SDHTFF in module SDR2. The TRIARG and TRAPRG elements are processed with the same equations as the TRMEM or QDMEM elements, except that the value of the "thickness"  $t$  is calculated as above.



# STRUCTURAL ELEMENT DESCRIPTIONS

## 4.87.11 The TRAPRG Element

### 4.87.11.1 Input Data for the TRAPRG Element

1. The ECPT/EST entries for the axisymmetric trapezoidal ring (TRAPRG) element are

| <u>Symbol</u>   | <u>Description</u>  |
|---|---|
| $SIL_1, SIL_2, SIL_3, SIL_4$  | Scalar index numbers for the four grid points.  |
| $\gamma$  | Material property orientation angle (degrees).  |
| Mat. I.D.   | Material property identification number.  |
| $\left. \begin{array}{l} N_1, X_1, Y_1, Z_1 \\ N_2, X_2, Y_2, Z_2 \\ N_3, X_3, Y_3, Z_3 \\ N_4, X_4, Y_4, Z_4 \end{array} \right\}$ | Local coordinate system number and location in basic coordinates of the four grid points. |
| $t_\mu$   | Element temperature for material properties   |

For this element  $Y_i$  must equal zero for  $i = 1, 2, 3$  and  $4$ , and we define:

$$\{R_s\} = \begin{Bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{Bmatrix}, \quad (1)$$

$$Z_s = \begin{Bmatrix} Z_{s1} \\ Z_{s2} \\ Z_{s3} \\ Z_{s4} \end{Bmatrix} = \begin{Bmatrix} Z_1 \\ Z_2 \\ Z_3 \\ Z_4 \end{Bmatrix}. \quad (2)$$



## MODULE FUNCTIONAL DESCRIPTIONS

### 2. Geometry Input

The location  $(X_i, Y_i, Z_i)$  and local coordinate system number  $(N_i)$  of each grid point are used to calculate the 3 by 3 global-to-basic coordinate system transformation matrices,  $[T_i]$ ,  $i = 1, 2, 3, 4$ .

### 3. Material Property Input

The material property input for the TRAPRG element is the same as that for the TRIARG element (see section 4.87.10.1).

#### 4.87.11.2 General Calculations

1. Local coordinate calculations are:

$$Z_{\min} = \text{minimum of } (Z_{s1}, Z_{s2}, Z_{s3}, Z_{s4}), \quad (3)$$

$$\{R_L\} = \{R_S\}, \quad (4)$$

$$\{Z_L\} = \{Z_S\} - \begin{pmatrix} Z_{\min} \\ Z_{\min} \\ Z_{\min} \\ Z_{\min} \end{pmatrix}. \quad (5)$$

Let  $R_{Li}$  and  $Z_{Li}$  be the  $i^{\text{th}}$  component of  $\{R_L\}$  and  $\{Z_L\}$  respectively. To insure the user has input his grid points in accordance with the restrictions set down in section 2 of the User's Manual, the following tests are made:

If  $R_{L1} \geq R_{L2}$  or  $R_{L4} \geq R_{L3}$  or  $Z_{L4} \leq Z_{L1}$ , then the user has violated the restriction that the grid points be ordered in a counterclockwise manner and a user fatal error occurs.

If  $|Z_{L1} - Z_{L2}| > .001$  or  $|Z_{L3} - Z_{L4}| > .001$ , then the restriction that the line joining grid points 1 and 2 and the line joining grid points 3 and 4 be parallel to the r-axis has been violated and a user fatal error occurs.

2. Test for a rectangle. Define:

$$R_{M14} = (R_{L1} + R_{L4})/2, \quad (6)$$

# STRUCTURAL ELEMENT DESCRIPTIONS

$$R_{M23} = (R_{L2} + R_{L3})/2. \quad (7)$$

$$\text{If } \left| \frac{R_{L1} - R_{L4}}{R_{M14}} \right| < 0.005 \text{ then } R_{L1} = R_{L4} = R_{M14}.$$

$$\text{If } \left| \frac{R_{L2} - R_{L3}}{R_{M23}} \right| < 0.005 \text{ then } R_{L2} = R_{L3} = R_{M23}.$$

If  $R_{L1} = R_{L4}$  and  $R_{L2} = R_{L3}$ , then the element is rectangular.

If  $R_{L1} = R_{L4} = 0$ , then the element is a core element.

3. Generate the transformation matrix from field coordinates to grid point degrees of freedom:

$$[\bar{H}] = \begin{bmatrix} 1 & R_{L1} & Z_{L1} & R_{L1}Z_{L1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & R_{L1} & Z_{L1} & R_{L1}Z_{L1} \\ 1 & R_{L2} & Z_{L1} & R_{L2}Z_{L1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & R_{L2} & R_{L1} & R_{L2}Z_{L1} \\ 1 & R_{L3} & Z_{L4} & R_{L3}Z_{L4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & R_{L3} & Z_{L4} & R_{L3}Z_{L4} \\ 1 & R_{L4} & Z_{L4} & R_{L4}Z_{L4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & R_{L4} & Z_{L4} & R_{L4}Z_{L4} \end{bmatrix} \quad (8)$$

$$[H] = [\bar{H}]^{-1} \quad (9)$$

4. Generate the transformation matrix from two to three degrees of freedom per point:

$$[\Gamma_{qs}] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

# MODULE FUNCTIONAL DESCRIPTIONS

5. If the element is a core element, then:

$$\begin{aligned} h_{1j} &= h_{3j} = 0 & j &= 1, 2, \dots, 8, \\ h_{i1} &= h_{i7} = 0 & i &= 1, 2, \dots, 8. \end{aligned} \quad (11)$$

where  $h_{ij}$  is an element of  $[H]$ .

## 4.87.11.3 Integral Calculations

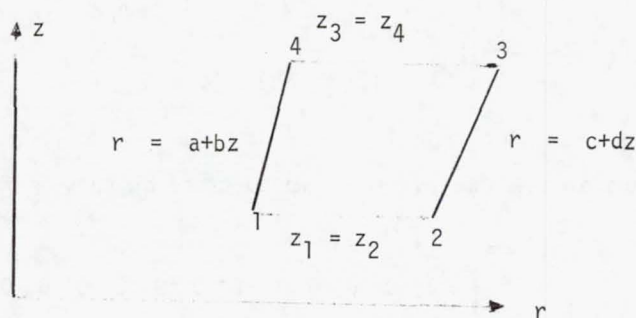
a. Compute the integrals over the cross-section of the trapezoid that are of the form:

$$I_{pq} = \iint_{rz} r^p z^q dr dz \quad (12)$$

for the values:

$$I_{-10}, I_{-11}, I_{-12}, I_{00}, I_{01}, I_{02}, I_{10}, I_{11}, I_{12}, I_{20}, I_{21}, I_{22}, I_{30}, I_{31}, I_{32}.$$

The limits of integration are chosen depending on the geometric shape of the trapezoid.



If the lines between points 1 and 4 and between points 2 and 3 are defined by the equations

$$r = a + bz, \quad (13)$$

$$r = c + dz, \quad (14)$$



# STRUCTURAL ELEMENT DESCRIPTIONS

respectively, then:

$$a = R_{L1} - \left( \frac{R_{L4} - R_{L1}}{Z_{L4} - Z_{L1}} \right) Z_{L1}, \quad (15)$$

$$b = \frac{R_{L4} - R_{L1}}{Z_{L4} - Z_{L1}}, \quad (16)$$

$$c = R_{L2} - \left( \frac{R_{L3} - R_{L2}}{Z_{L3} - Z_{L2}} \right) Z_{L2}, \quad (17)$$

$$d = \frac{R_{L3} - R_{L2}}{Z_{L3} - Z_{L2}}. \quad (18)$$

In general, the integration takes the form:

$$I_{Pq} = \int_{z_1}^{z_4} \int_{a+bz}^{c+dz} r^P z^Q dr dz. \quad (19)$$

For the case with the side  $r = c + dz$  parallel to the axis of symmetry (the  $z$  axis) we have:

$$I_{Pq} = \int_{z_1}^{z_4} \int_{a+bz}^c r^P z^Q dr dz. \quad (20)$$

For the case with the side  $r = a + bz$  parallel to the axis of symmetry we have:

$$I_{Pq} = \int_{z_1}^{z_4} \int_a^{c+dz} r^P z^Q dr dz. \quad (21)$$

And finally for the rectangle, the integration takes the form

# MODULE FUNCTIONAL DESCRIPTIONS

$$I_{pq} = \int_{z_1}^{z_4} \int_a^c r^p z^q dr dz. \quad (22)$$

## 4.87.11.4 Elastic Constants Matrix Calculation

The elastic constants matrix in element coordinates,  $[E_g]$ , for the TRAPRG element is calculated identically to the elastic constants matrix for the TRIARG element (see section 4.87.10.4).

## 4.87.11.5 Stiffness Matrix Generation (Subroutine KTRAPR of Module SMA1)

1. Generate the terms of the symmetric element stiffness matrix in field coordinates as shown in Table 2. Each term must be multiplied by  $2\pi$  to form  $[\tilde{K}]$ .
2. Transform the element stiffness matrix from field coordinates to grid point degrees of freedom:

$$[\tilde{K}] = [H]^T [\tilde{K}] [H]. \quad (23)$$

3. Transform the element stiffness matrix from two to three degrees of freedom per point:

$$[K] = [\Gamma_{qs}]^T [\tilde{K}] [\Gamma_{qs}]. \quad (24)$$

4. The 3 by 3 partitions  $[K_{pj}]$  of  $[K]$  corresponding to the pivot point p are transformed:

$$[K_{pj}^3] = [T_p]^T [K_{pj}^3] [T_j], \quad j = 1, 2, 3, 4. \quad (25)$$

5. Finally, these 3 by 3 partitions are expanded to 6 by 6 partitions:

$$[K_{pj}] = \begin{bmatrix} K_{pj}^3 & 0 \\ 0 & 0 \end{bmatrix}. \quad (26)$$

# STRUCTURAL ELEMENT DESCRIPTIONS

Table 2. Elements of the 8 by 8 Symmetric Stiffness Matrix for the TRAPRG Element.

|       | Col. 1                  | Col. 2                          | Col. 3                                      | Col. 4  |
|-------|-------------------------|---------------------------------|---|---|
| Row 1 | $E_{22}I_{-10}$         |                                 |   |   |
| Row 2 | $(E_{22}+E_{12})I_{00}$ | $(E_{11}+2E_{12}+E_{22})I_{10}$ |   |   |
| Row 3 | $E_{22}I_{-11}$         | $(E_{12}+E_{22})I_{01}$         | $E_{22}I_{-12}+E_{44}I_{10}$                |   |
| Row 4 | $(E_{12}+E_{22})I_{01}$ | $(E_{11}+2E_{12}+E_{22})I_{11}$ | $(E_{12}+E_{22})I_{02}$<br>+ $E_{44}I_{20}$ | $(E_{11}+2E_{12}+E_{22})I_{12}$<br>+ $E_{44}I_{30}$ |
| Row 5 | 0                       | 0                               | 0   | 0   |
| Row 6 | 0                       | 0                               | $E_{44}I_{10}$                              | $E_{44}I_{20}$                                      |
| Row 7 | $E_{32}I_{00}$          | $(E_{13}+E_{23})I_{10}$         | $E_{23}I_{01}$                              | $(E_{13}+E_{23})I_{11}$                             |
| Row 8 | $E_{32}I_{10}$          | $(E_{31}+E_{32})I_{20}$         | $(E_{23}+E_{44})I_{11}$                     | $(E_{13}+E_{23}+E_{44})I_{21}$                      |

|       | Col. 5 | Col. 6         | Col. 7         | Col. 8                      |
|-------|--------|----------------|----------------|-----------------------------|
| Row 5 | 0      |                |                |                             |
| Row 6 | 0      | $E_{44}I_{10}$ |                |                             |
| Row 7 | 0      | 0              | $E_{33}I_{10}$ |                             |
| Row 8 | 0      | $E_{44}I_{11}$ | $E_{33}I_{20}$ | $E_{33}I_{30}+E_{44}I_{12}$ |



# MODULE FUNCTIONAL DESCRIPTIONS

## 4.87.11.6 Mass Matrix Calculation (Subroutine MTRAPR of Module SMA2)

1. Form the coupled mass matrix in field coordinates:

$$[\tilde{M}] = 2\pi \begin{bmatrix} M_1 I_{10} & & & & & & & \\ M_1 I_{20} & M_1 I_{30} & & & & & & \\ & & \text{Symmetric} & & & & & \\ M_1 I_{11} & M_1 I_{21} & M_1 I_{12} & & & & & \\ M_1 I_{21} & M_1 I_{31} & M_1 I_{22} & M_1 I_{32} & & & & \\ 0 & 0 & 0 & 0 & M_2 I_{10} & & & \\ 0 & 0 & 0 & 0 & M_2 I_{20} & M_2 I_{30} & & \\ 0 & 0 & 0 & 0 & M_2 I_{11} & M_2 I_{21} & M_2 I_{12} & \\ 0 & 0 & 0 & 0 & M_2 I_{21} & M_2 I_{31} & M_2 I_{22} & M_2 I_{32} \end{bmatrix} \cdot \quad (27)$$

8x8

where

$$M_1 = M_2 = \rho \cdot \quad (28)$$

2. Transform the mass matrix to grid point degrees of freedom:

$$[\bar{M}] = [H]^T [\tilde{M}] [H]. \quad (29)$$

3. Transform the mass matrix from two to three degrees of freedom per point:

$$[M] = [\Gamma_{qs}] [\bar{M}] [\Gamma_{qs}]. \quad (30)$$

4. The 6 by 6 partitions,  $[M_{pj}]$ , are calculated as in Equations 25 and 26.

# STRUCTURAL ELEMENT DESCRIPTIONS

## 4.87.11.7 Thermal Load Calculations (Subroutine TTRAPR of Module SSG1)

1. Form the temperature gradient vector:

$$\{\Delta T\} = \{T\} - \begin{pmatrix} T_o \\ T_o \\ T_o \\ T_o \end{pmatrix}, \quad (31)$$

where  $\{T\}$  is the vector of loading temperatures at the grid points.

2. Form the thermal expansion coefficient vector:

$$\{\alpha\} = \begin{pmatrix} \alpha_r \\ \alpha_\theta \\ \alpha_z \\ 0 \end{pmatrix}. \quad (32)$$

3. Multiply the elastic constants matrix by the thermal expansion coefficient vector:

$$\{AB\} = [E_g]\{\alpha\}. \quad (33)$$

4. Form the  $[Q]$  matrix:

$$[Q] = \begin{bmatrix} AB_2 I_{00} & AB_2 I_{10} & AB_2 I_{01} & AB_2 I_{11} \\ (AB_1 + AB_2) I_{10} & (AB_1 + AB_2) I_{20} & (AB_1 + AB_2) I_{11} & (AB_1 + AB_2) I_{21} \\ AB_2 I_{01} & AB_2 I_{11} & AB_2 I_{02} & AB_2 I_{12} \\ (AB_1 + AB_2) I_{11} & (AB_1 + AB_2) I_{21} & (AB_1 + AB_2) I_{12} & (AB_1 + AB_2) I_{22} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ AB_3 I_{10} & AB_3 I_{20} & AB_3 I_{11} & AB_3 I_{21} \\ AB_3 I_{20} & AB_3 I_{30} & AB_3 I_{21} & AB_3 I_{31} \end{bmatrix} \cdot \quad (34)$$

8x4

# MODULE FUNCTIONAL DESCRIPTIONS

5. Partition the transformation matrix  $[H]$  to form  $[H']$ :

$$[H'] = [h'_{ij}] \quad 4 \times 4, \quad (35)$$

where

$$h'_{ij} = h_{ik} \quad \text{for } i = 1, 2, 4; \quad j = 1, 2, \dots, 4; \quad \text{and } k = 2j-1$$

and  $h_{ik}$  are the elements of  $[H]$ .

6. Compute the thermal load in field coordinates:

$$\{\tilde{F}_T\} = 2\pi[Q][H']\{\Delta T\}. \quad (36)$$

7. Transform the thermal load to grid point degrees of freedom:

$$\{\bar{F}_T\} = [H]^T\{\tilde{F}_T\}. \quad (37)$$

8. Transform the thermal load from two to three degrees of freedom per point:

$$\{F_T\} = [\Gamma_{qs}]^T\{\bar{F}_T\}. \quad (38)$$

9. Each partition,  $\{F_T^3\}$ , of length 3 of  $\{F_T\}$  is transformed to global coordinates by:

$$\{F_T^3\}_g = [T_i]^T\{F_T^3\}. \quad (39)$$

10. These vectors are added to the overall load vector,  $\{P_g\}$ .

## 4.87.11.8 Element Force and Stress Calculations (Subroutines STRAP1 and STRAP2 of Module SDR2).

Element stress and force data items are calculated in two phases. The first phase (subroutine STRAP1) calculates the element stiffness and stress matrices. The second phase (subroutine STRAP2) calculates the element forces and stresses from the various subcase displacement vectors.



# STRUCTURAL ELEMENT DESCRIPTIONS

Phase 1 calculations are as follows:

1. Form the element stiffness matrix,  $[K]$ , as in section 4.87.11.5.
2. Define a fifth "grid point" to be the average of the other four points:

$$R_{L5} = \frac{1}{4}(R_{L1} + R_{L2} + R_{L3} + R_{L4}), \quad (40)$$

$$Z_{L5} = \frac{1}{4}(Z_{L1} + Z_{L2} + Z_{L3} + Z_{L4}). \quad (41)$$

3. Form the matrices  $[D^{(1)}], [D^{(2)}], [D^{(3)}], [D^{(4)}], [D^{(5)}]$

where

$$[D^{(i)}] = \begin{bmatrix} 0 & 1 & 0 & Z_{Li} & 0 & 0 & 0 & 0 \\ \frac{1}{R_{Li}} & 1 & \frac{Z_{Li}}{R_{Li}} & Z_{Li} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & R_{Li} \\ 0 & 0 & 1 & R_{Li} & 0 & 1 & 0 & Z_{Li} \end{bmatrix}_{4 \times 8}, \quad (42)$$

where  $i = 1$  to 5 denotes the five grid points.

4. Compute the stress matrices for each of the five grid points in field coordinates:

$$[\tilde{S}^{(i)}] = [E_g][D^{(i)}]. \quad (43)$$

# MODULE FUNCTIONAL DESCRIPTIONS

5. Transform each stress matrix to grid point degrees of freedom:

$$[\bar{S}^{(i)}] = [\tilde{S}^{(i)}][H]. \quad (44)$$

6. Transform each stress matrix from two to three degrees of freedom per point:

$$[S^{(i)}] = [\bar{S}^{(i)}][T_{qs}]. \quad (45)$$

7. Form the master stress matrix:

$$[S] = \begin{bmatrix} S^{(1)} \\ S^{(2)} \\ S^{(3)} \\ S^{(4)} \\ S^{(5)} \end{bmatrix} \cdot 20 \times 12 \quad (46)$$

8. Transform the stress matrix from basic to local coordinates:

$$[S_\ell] = [S][T_{1234}], \quad (47)$$

where

$$[T_{1234}] = \begin{bmatrix} T_1 & 0 & 0 & 0 \\ 0 & T_2 & 0 & 0 \\ 0 & 0 & T_3 & 0 \\ 0 & 0 & 0 & T_4 \end{bmatrix} \cdot \quad (48)$$

9. Compute the thermal stress vector:

$$\{T_s\} = [E_g]\{\alpha\}. \quad (49)$$

Phase 2 calculations are as follows:

1. Extract the displacement vector,  $\{\Delta\}$ , at the three translation coordinates of each of the four grid points from the global displacement vector.

# STRUCTURAL ELEMENT DESCRIPTIONS

2. Calculate the element forces:

$$\{P\} = [K]\{\Delta\} . \quad (50)$$

3. Calculate the element stresses:

$$\{\sigma\} = [S_{\ell}]\{\Delta\} - T_d \{T_S\} , \quad (51)$$

where

$$T_d = \begin{cases} T_i - T_0, & \text{if } i \neq 5 \text{ (} T_i \text{ is the temperature at the } i^{\text{th}} \text{ point)} \\ T_{\text{avg}} - T_0, & \text{if } i = 5 \text{ (} T_{\text{avg}} \text{ is the average temperature over the four grid points)} \end{cases} \quad (52)$$

## 4.87.11.9 Thermal Analysis Calculations for the TRAPRG Element (Subroutine HRING by Module SMA1)

The calculations are described in the preceding description for the TRIARG element; see Section 4.87.10.9.



## MODULE FUNCTIONAL DESCRIPTIONS

### 4.87.12 The TØRDRG Element

#### 4.87.12.1 Input Data for the TØRDRG Element

1. The ECPT/EST entries for the axisymmetric toroidal ring (TØRDRG) element are:

| <u>Symbol</u>  | <u>Description</u>   |
|--|--|
| $SIL_1, SIL_2$   | Scalar index numbers for the two grid points   |
| $\alpha_1, \alpha_2$   | Angles of curvature at the two grid points (degrees)                                 |
| $\gamma$   | Material property orientation angle (currently not used)                             |
| $H_m, H_f$   | Membrane and flexure thickness   |
| $\left. \begin{matrix} N_1, X_1, Y_1, Z_1 \\ N_2, X_2, Y_2, Z_2 \end{matrix} \right\}$ | Local coordinate system number and location in basic coordinates of the grid points. |
| $t_\mu$  | Element temperature for material properties  |

For this element  $Y_i$  must equal zero for  $i = 1$  and  $2$ , and we define:

$$\{R\} = \begin{Bmatrix} X_1 \\ X_2 \end{Bmatrix} , \quad (1)$$

$$\{Z\} = \begin{Bmatrix} Z_1 \\ Z_2 \end{Bmatrix} . \quad (2)$$

#### 2. Coordinate system data

The location  $(X_i, Y_i, Z_i)$  and local coordinate system number  $(N_i)$  of each grid point are used to calculate the 3 by 3 global-to-basic coordinate system transformation matrices,  $[T_i]$ ,  $i = 1, 2$ .

#### 3. Material data

The material property input for the TØRDRG element is the same as that for TRIARG element (see section 4.87.10.1) with the following notational changes:

# STRUCTURAL ELEMENT DESCRIPTIONS

$$E_p = E_r, \quad E_T = E_\theta, \quad \nu_{pT} = \nu_{r\theta}$$

$$\{ALF\} = \begin{Bmatrix} \alpha_r \\ \alpha_\theta \end{Bmatrix}$$

## 4.87.12.2 General Calculations

1. Compute the following constants used in stiffness matrix generation:

$$C = E_p/E_T, \quad (3)$$

$$D_M = E_p H_m / (C - \nu_{pT}^2), \quad (4)$$

$$D_B = E_p H_f^3 / (12 (C - \nu_{pT}^2)). \quad (5)$$

2. Determine if the element is a toroidal ring, a conical ring, a cylindrical ring, or a shell cap:

- (1) if  $\alpha_1 \neq \alpha_2$ , then the element is a toroidal ring.
- (2) if  $\alpha_1 = \alpha_2 = 90^\circ$ , then the element is a cylindrical ring.
- (3) if  $\alpha_1 = \alpha_2 \neq 90^\circ$ , then the element is a conical ring.
- (4) if  $\alpha_1 = 0$ , then the element is a shell cap.

3. Compute the local coordinate constants for a toroidal element:

$$\phi_\beta = \alpha_2 - \alpha_1, \quad (6)$$

$$R_p = \frac{[(R_2 - R_1)^2 + (Z_2 - Z_1)^2]^{1/2}}{2 \sin(\frac{\phi_\beta}{2})}, \quad (7)$$

$$\lambda_1 = \frac{1}{R_p}, \quad (8)$$

# MODULE FUNCTIONAL DESCRIPTIONS

$$S = (\alpha_2 - \alpha_1)R_p, \quad (9)$$

$$B_\beta = R_1 + R_p [\sin(\alpha_1 + \frac{\phi_\beta}{2}) - \sin(\alpha_1)], \quad (10)$$

$$R_T = \frac{B_\beta}{\sin(\alpha_1 + \frac{\phi_\beta}{2})}, \quad (11)$$

$$\psi_1 = \cos(\alpha_1 + \frac{\phi_\beta}{2}), \quad (12)$$

$$\psi_2 = -\frac{\sin(\alpha_1 + \frac{\phi_\beta}{2})}{R_p}. \quad (13)$$

4. Compute the local coordinate constants for a conical or cylindrical ring

$$S = [(R_2 - R_1)^2 + (Z_2 - Z_1)^2]^{1/2}, \quad (14)$$

$$B_\beta = R_1 + \frac{S \cos \alpha_1}{2}, \quad (15)$$

$$R_T = \frac{B_\beta}{\sin \alpha_1}, \quad (16)$$

$$R_p = 0, \quad (17)$$

$$\lambda_1 = 0, \quad (18)$$

$$\psi_1 = \cos \alpha_1, \quad (19)$$

$$\psi_2 = 0. \quad (20)$$



# STRUCTURAL ELEMENT DESCRIPTIONS

5. Generate the transformation matrix from field coordinates to grid point degrees of freedom:

$$[r_{\beta q}] = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-10}{s^3} & \frac{-6}{s^2} & 0 & \frac{-3}{2s} & 0 & 0 & \frac{10}{s^3} & \frac{-4}{s^2} & 0 & \frac{1}{2s} \\ 0 & 0 & \frac{15}{s^4} & \frac{8}{s^3} & 0 & \frac{3}{2s^2} & 0 & 0 & \frac{-15}{s^4} & \frac{7}{s^3} & 0 & \frac{-1}{s^2} \\ 0 & 0 & \frac{-6}{s^5} & \frac{-3}{s^4} & 0 & \frac{-1}{2s^3} & 0 & 0 & \frac{6}{s^5} & \frac{-3}{s^4} & 0 & \frac{1}{2s^3} \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{-3}{s^2} & 0 & 0 & 0 & \frac{-2}{s} & 0 & \frac{3}{s^2} & 0 & 0 & 0 & \frac{-1}{s} & 0 \\ \frac{2}{s^3} & 0 & 0 & 0 & \frac{1}{s^2} & 0 & \frac{-2}{s^3} & 0 & 0 & 0 & \frac{1}{s^2} & 0 \end{bmatrix} \cdot (21) \quad 10 \times 12$$

# MODULE FUNCTIONAL DESCRIPTIONS

6. Generate the transformation matrix from local to system coordinates:

$$[r_{rs}] = \begin{bmatrix} \cos \alpha_1 & 0 & -\sin \alpha_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \sin \alpha_1 & 0 & \cos \alpha_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cos \alpha_2 & 0 & -\sin \alpha_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sin \alpha_2 & 0 & \cos \alpha_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad 12 \times 12 \quad (22)$$

7. Rearrange the transformation matrix  $[r_{\beta q}]$  such that the membrane and flexure terms are reversed.

$$[\bar{r}_{\beta q}] = \begin{bmatrix} r_{\beta q}^{(2)} \\ r_{\beta q}^{(1)} \end{bmatrix} \quad 10 \times 12, \quad (23)$$

where  $[r_{\beta q}^{(1)}]$  is the first six rows of  $[r_{\beta q}]$  and  $[r_{\beta q}^{(2)}]$  is the last four rows of  $[r_{\beta q}]$ .

## 4.87.12.3 Integral Calculations

The method used to compute the integrals depends on the geometric shape of the element.

### 1. Toroidal ring - basic integrals

There are six basic integrals to be computed, of which the first three can best be evaluated by series expansion, but the last three require numerical integration.

# STRUCTURAL ELEMENT DESCRIPTIONS

$$I_1^j = R_p^{j+1} \int_0^{\phi_\beta} \phi^j d\phi = \frac{R_p^{j+1} \phi_\beta^{j+1}}{j+1}, \quad (24)$$

$$I_2^j = R_p^{j+1} \int_0^{\phi_\beta} \phi^j \sin \phi d\phi = R_p^{j+1} \sum_{i=0}^{\infty} \frac{(-1)^i \phi_\beta^{j+1+2i+1}}{(j+1+2i+1)(2i+1)!}, \quad (25)$$

$$I_3^j = R_p^{j+1} \int_0^{\phi_\beta} \phi^j \cos \phi d\phi = R_p^{j+1} \sum_{i=0}^{\infty} \frac{(-1)^i \phi_\beta^{j+1+2i}}{(j+1+2i)(2i)!}, \quad (26)$$

$$I_4^j = R_p^{j+1} \int_0^{\phi_\beta} \frac{\phi^j \sin^2 \phi}{B} d\phi, \quad (27)$$

$$I_5^j = R_p^{j+1} \int_0^{\phi_\beta} \frac{\phi^j 2 \sin \phi \cos \phi}{B} d\phi, \quad (28)$$

$$I_6^j = R_p^{j+1} \int_0^{\phi_\beta} \frac{\phi^j \cos^2 \phi}{B} d\phi, \quad (29)$$

where

$$B = R_1 - R_p \sin \alpha_1 + R_p \sin \alpha_1 \cos \phi + R_p \cos \alpha_1 \sin \phi, \quad (30)$$

and  $j = 0, 1, \dots, 10$ .

The summations in  $I_2^j$  and  $I_3^j$  are carried out until the truncation error is insignificant.



2. Toroidal ring - required integrals

The actual integrals required can now be defined in terms of basic integrals.

$$\delta_1^j = (R_1 - R_p \sin \alpha_1) I_1^j + R_p \cos \alpha_1 I_2^j + R_p \sin \alpha_1 I_3^j, \quad (31)$$

$$\delta_2^j = \cos \alpha_1 I_2^j + \sin \alpha_1 I_3^j, \quad (32)$$

$$\delta_3^j = \cos^2 \alpha_1 I_4^j + \sin \alpha_1 \cos \alpha_1 I_5^j + \sin^2 \alpha_1 I_6^j, \quad (33)$$

$$\delta_4^j = \cos \alpha_1 I_3^j - \sin \alpha_1 I_2^j, \quad (34)$$

$$\delta_5^j = \sin \alpha_1 \cos \alpha_1 (I_6^j - I_4^j) + \frac{1}{2} (1 - 2 \sin^2 \alpha_1) I_5^j, \quad (35)$$

$$\delta_6^j = \cos^2 \alpha_1 I_6^j - \sin \alpha_1 \cos \alpha_1 I_5^j + \sin^2 \alpha_1 I_4^j, \quad (36)$$

3. Conic ring - basic integrals

$$I_1^j = \int_0^S \xi^j d\xi = \frac{S^{j+1}}{j+1}, \quad j = 0, 1, \dots, 10 \quad (37)$$

$$I_2^j = \int_0^S \left( \frac{\xi^j}{R_1 + \xi \cos \alpha_1} \right) d\xi, \quad j = 0, 1, \dots, 10 \quad (38)$$

$$I_2^0 = \frac{1}{\cos \alpha_1} \ln \left( \frac{R_1 + S \cos \alpha_1}{R_1} \right), \quad (39)$$

$$I_2^1 = \frac{1}{\cos \alpha_1} \left[ S - \frac{R_1}{\cos \alpha_1} \ln \left( \frac{R_1 + S \cos \alpha_1}{R_1} \right) \right], \quad (40)$$

$$I_2^j = \frac{S^{j+1}}{R_1} \sum_{i=0}^{\infty} \frac{(-1)^i \left[ \frac{S \cos \alpha_1}{R_1} \right]^i}{(j+1+i)}, \quad j = 2, 3, \dots, 10. \quad (41)$$

# STRUCTURAL ELEMENT DESCRIPTIONS

## 4. Conic ring - required integrals

$$\delta_1^j = R_1 I_1^j + \psi_1 I_1^{j+1}, \quad (42)$$

$$\delta_2^j = \sin \alpha_1 I_1^j, \quad (43)$$

$$\delta_3^j = \sin^2 \alpha_1 I_2^j, \quad (44)$$

$$\delta_4^j = \psi_1 I_1^j, \quad j = 0, 1, \dots, 10 \quad (45)$$

$$\delta_5^j = \sin \alpha_1 \psi_1 I_2^j, \quad (46)$$

$$\delta_6^j = \psi_1^2 I_2^j, \quad (47)$$

## 5. Cylindrical ring - basic integrals

$$I_1^j = \int_0^S \xi^j d\xi = \frac{S^{j+1}}{j+1}, \quad j = 0, 1, \dots, 10 \quad (48)$$

$$I_2^j = \int_0^S \frac{\xi^j}{R_1} d\xi = \frac{1}{R_1} \left( \frac{S^{j+1}}{j+1} \right), \quad j = 0, 1, \dots, 10 \quad (49)$$

## 6. Cylindrical ring - required integrals

$$\delta_1^j = R_1 I_1^j + \psi_1 I_1^{j+1}, \quad (50)$$

$$\delta_2^j = \sin \alpha_1 I_1^j, \quad (51)$$

$$\delta_3^j = \sin^2 \alpha_1 I_2^j, \quad (52)$$

$$\delta_4^j = \delta_5^j = \delta_6^j = 0, \quad j = 0, 1, \dots, 10 \quad (53)$$

# MODULE FUNCTIONAL DESCRIPTIONS

## 4.87.12.4 Elastic Constants Matrix Calculations

Form elastic constants matrix

$$[E] = \frac{1}{\left(1 - \frac{E_T}{E_P} \nu_{PT}^2\right)} \begin{bmatrix} E_P & E_T \nu_{PT} \\ E_T \nu_{PT} & E_T \end{bmatrix} \cdot 2 \times 2 \quad (54)$$

## 4.87.12.5 Stiffness Matrix Calculations (Subroutine KTØRDR of Module SMA1)

1. Define the constants

$$A = R_P, \quad (55)$$

$$V = \nu_{PT}, \quad (56)$$

$$C = E_P/E_T. \quad (57)$$

2. Form the stiffness matrix terms in field coordinates as shown in Tables 3, 4 and 5.

3. Transform the stiffness matrix from field coordinates to grid point degrees of freedom:

$$[\bar{K}] = [r_{\beta q}]^T [\tilde{K}] [r_{\beta q}]. \quad (58)$$

4. Transform the stiffness matrix from local to system coordinates:

$$[K] = [r_{rs}]^T [\bar{K}] [r_{rs}]. \quad (59)$$

5. The global-to-basic coordinate system transformation matrices  $[T_i]$  are expanded to 6 by 6 matrices:

$$[T_i^6] = \begin{bmatrix} T_i & \begin{matrix} | & 0 \end{matrix} \\ \hline 0 & \begin{matrix} | & I \end{matrix} \end{bmatrix}, \quad i = 1, 2. \quad (60)$$



# STRUCTURAL ELEMENT DESCRIPTIONS

Table 3. Columns 1, 2 and 3 of the Symmetric 10 by 10 Stiffness Matrix for the TØRDRG Element.

|        | Column 1   | Column 2  | Column 3   |
|--------|--|---|--|
| Row 1  | $D_M(\frac{C}{A^2} \delta_1^0 + \frac{2V}{A} \delta_2^0 + \delta_3^0)$               |   |  |
| Row 2  | $D_M(\frac{C}{A^2} \delta_1^1 + \frac{2V}{A} \delta_2^1 + \delta_3^1)$               | $D_B \delta_6^0 + D_M(\frac{C}{A^2} \delta_1^2 + \frac{2V}{A} \delta_2^2 + \delta_3^2)$                     |  |
| Row 3  | $D_M(\frac{C}{A^2} \delta_1^2 + \frac{2V}{A} \delta_2^2 + \delta_3^2)$               | $D_B(2V \delta_4^0 + 2 \delta_6^1) + D_M(\frac{C}{A^2} \delta_1^3 + \frac{2V}{A} \delta_2^3 + \delta_3^3)$  | $D_B 4(C \delta_1^0 + 2V \delta_4^1 + \delta_6^2) + D_M(\frac{C}{A^2} \delta_1^4 + \frac{2V}{A} \delta_2^4 + \delta_3^4)$      |
| Row 4  | $D_M(\frac{C}{A^2} \delta_1^3 + \frac{2V}{A} \delta_2^3 + \delta_3^3)$               | $D_B(6V \delta_4^1 + 3 \delta_6^2) + D_M(\frac{C}{A^2} \delta_1^4 + \frac{2V}{A} \delta_2^4 + \delta_3^4)$  | $D_B 2(2C \delta_1^1 + 3V \delta_4^2 + \delta_6^3) + D_M(\frac{C}{A^2} \delta_1^5 + \frac{2V}{A} \delta_2^5 + \delta_3^5)$     |
| Row 5  | $D_M(\frac{C}{A^2} \delta_1^4 + \frac{2V}{A} \delta_2^4 + \delta_3^4)$               | $D_B(12V \delta_4^2 + 4 \delta_6^3) + D_M(\frac{C}{A^2} \delta_1^5 + \frac{2V}{A} \delta_2^5 + \delta_3^5)$ | $D_B 2(12C \delta_1^2 + 16V \delta_4^3 + 4 \delta_6^4) + D_M(\frac{C}{A^2} \delta_1^6 + \frac{2V}{A} \delta_2^6 + \delta_3^6)$ |
| Row 6  | $D_M(\frac{C}{A^2} \delta_1^5 + \frac{2V}{A} \delta_2^5 + \delta_3^5)$               | $D_B(20V \delta_4^3 + 5 \delta_6^4) + D_M(\frac{C}{A^2} \delta_1^6 + \frac{2V}{A} \delta_2^6 + \delta_3^6)$ | $D_B 10(4C \delta_1^3 + 5V \delta_4^4 + \delta_6^5) + D_M(\frac{C}{A^2} \delta_1^7 + \frac{2V}{A} \delta_2^7 + \delta_3^7)$    |
| Row 7  | $D_M(\frac{V}{A} \delta_4^0 + \delta_5^0)$   | $D_M(\frac{V}{A} \delta_4^1 + \delta_5^1)$  | $D_M(\frac{V}{A} \delta_4^2 + \delta_5^2)$   |
| Row 8  | $D_M(\frac{C}{A} \delta_1^0 + \frac{V}{A} \delta_4^1 + V \delta_2^0 + \delta_5^1)$   | $D_M(\frac{C}{A} \delta_1^1 + \frac{V}{A} \delta_4^2 + V \delta_2^1 + \delta_5^2)$                          | $D_M(\frac{C}{A} \delta_1^2 + \frac{V}{A} \delta_4^3 + V \delta_2^2 + \delta_5^3)$   |
| Row 9  | $D_M(\frac{2C}{A} \delta_1^1 + \frac{V}{A} \delta_4^2 + 2V \delta_2^1 + \delta_5^2)$ | $D_M(2\frac{C}{A} \delta_1^2 + \frac{V}{A} \delta_4^3 + 2V \delta_2^2 + \delta_5^3)$                        | $D_M(2\frac{C}{A} \delta_1^3 + \frac{V}{A} \delta_4^4 + 2V \delta_2^3 + \delta_5^4)$   |
| Row 10 | $D_M(\frac{3C}{A} \delta_1^2 + \frac{V}{A} \delta_4^3 + 3V \delta_2^2 + \delta_5^3)$ | $D_M(\frac{3C}{A} \delta_1^3 + \frac{V}{A} \delta_4^4 + 3V \delta_2^3 + \delta_5^4)$                        | $D_M(\frac{3C}{A} \delta_1^4 + \frac{V}{A} \delta_4^5 + 3V \delta_2^4 + \delta_5^5)$   |

MODULE FUNCTIONAL DESCRIPTIONS

Table 4. Columns 4, 5, and 6 of the Symmetric 10 by 10 Stiffness Matrix for the TØRDRG Element.

|        | Column 4   | Column 5  | Column 6   |
|--------|--|---|--|
| Row 4  | $D_B 9(4C\delta_1^2 + 4V\delta_4^3 + \delta_6^4)$<br>$+ D_M(\frac{C}{A^2} \delta_1^6 + \frac{2V}{A} \delta_2^6 + \delta_3^6)$  |   |  |
| Row 5  | $D_B 12(6C\delta_1^3 + 5V\delta_4^4 + \delta_6^5)$<br>$+ D_M(\frac{C}{A^2} \delta_1^7 + \frac{2V}{A} \delta_2^7 + \delta_3^7)$ | $D_B 16(9C\delta_1^4 + 6V\delta_4^5 + \delta_6^6)$<br>$+ D_M(\frac{C}{A^2} \delta_1^8 + \frac{2V}{A} \delta_2^8 + \delta_3^8)$  |  |
| Row 6  | $D_B 15(8C\delta_1^4 + 6V\delta_4^5 + \delta_6^6)$<br>$+ D_M(\frac{C}{A^2} \delta_1^8 + \frac{2V}{A} \delta_2^8 + \delta_3^8)$ | $D_B 20(12C\delta_1^5 + 7V\delta_4^6 + \delta_6^7)$<br>$+ D_M(\frac{C}{A^2} \delta_1^9 + \frac{2V}{A} \delta_2^9 + \delta_3^9)$ | $D_E 25(16C\delta_1^6 + 8V\delta_4^7 + \delta_6^8)$<br>$+ D_M(\frac{C}{A^2} \delta_1^{10} + \frac{2V}{A} \delta_2^{10} + \delta_3^{10})$ |
| Row 7  | $D_M(\frac{V}{A} \delta_4^3 + \delta_5^3)$   | $D_M(\frac{V}{A} \delta_4^4 + \delta_5^4)$  | $D_M(\frac{V}{A} \delta_4^5 + \delta_5^5)$   |
| Row 8  | $D_M(\frac{C}{A} \delta_1^3 + \frac{V}{A} \delta_4^4$<br>$+ V\delta_2^3 + \delta_5^4)$   | $D_M(\frac{C}{A} \delta_1^4 + \frac{V}{A} \delta_4^5$<br>$+ V\delta_2^4 + \delta_5^5)$  | $D_M(\frac{C}{A} \delta_1^5 + \frac{V}{A} \delta_4^6$<br>$+ V\delta_2^5 + \delta_5^6)$   |
| Row 9  | $D_M(\frac{2C}{A} \delta_1^4 + \frac{V}{A} \delta_4^5$<br>$+ 2V\delta_2^4 + \delta_5^5)$                                       | $D_M(\frac{2C}{A} \delta_1^5 + \frac{V}{A} \delta_4^6$<br>$+ 2V\delta_2^5 + \delta_5^6)$  | $D_M(\frac{2C}{A} \delta_1^6 + \frac{V}{A} \delta_4^7$<br>$+ 2V\delta_2^6 + \delta_5^7)$   |
| Row 10 | $D_M(\frac{3C}{A} \delta_1^5 + \frac{V}{A} \delta_4^5$<br>$+ 3V\delta_2^5 + \delta_5^5)$                                       | $D_M(\frac{3C}{A} \delta_1^6 + \frac{V}{A} \delta_4^7$<br>$+ 3V\delta_2^6 + \delta_5^7)$  | $D_M(\frac{3C}{A} \delta_1^7 + \frac{V}{A} \delta_4^8$<br>$+ 3V\delta_2^7 + \delta_5^8)$   |

# STRUCTURAL ELEMENT DESCRIPTIONS

Table 5. Columns 7, 8, 9, and 10 of the Symmetric 10 by 10 Stiffness Matrix for the TØRDRG Element.

|        | Col. 7                           | Col. 8  | Col. 9  | Col. 10   |
|--------|----------------------------------|---|---|---|
| Row 7  | $D_M \delta_6^0$                 |   |   |   |
| Row 8  | $D_M(V\delta_4^0 + \delta_6^1)$  | $D_M(C\delta_1^0 + 2V\delta_4^1 + \delta_6^2)$  |   |   |
| Row 9  | $D_M(2V\delta_4^1 + \delta_6^2)$ | $D_M(2C\delta_1^1 + 3V\delta_4^2 + \delta_6^3)$ | $D_M(4C\delta_1^2 + 4V\delta_4^3 + \delta_6^4)$ |   |
| Row 10 | $D_M(3V\delta_4^2 + \delta_6^3)$ | $D_M(3C\delta_1^2 + 4V\delta_4^3 + \delta_6^4)$ | $D_M(6C\delta_1^3 + 5V\delta_4^4 + \delta_6^5)$ | $D_M(9C\delta_1^4 + 6V\delta_4^5 + \delta_6^6)$ |



# MODULE FUNCTIONAL DESCRIPTIONS

6. The stiffness matrix,  $[K]$ , is partitioned into 6 by 6 matrices:

$$[K] \Rightarrow \begin{bmatrix} K_{11}^6 & K_{12}^6 \\ K_{21}^6 & K_{22}^6 \end{bmatrix} \quad (61)$$

7. These 6x6 partitions are transformed to local coordinates:

$$[K_{pj}^6]_L = [T_p^6]^T [K_{pj}^6] [T_j^6], \quad (62)$$

where  $j = 1, 2$ , and  $p$  is the pivot point,  $p = 1$  or  $2$ .

# STRUCTURAL ELEMENT DESCRIPTIONS

## 4.87.12.6 Mass Matrix Calculations (Subroutine MTØRDR of Module SMA2)

1. Form the mass matrix in field coordinates:

$$[\tilde{M}] = 2\pi\rho H_m \begin{bmatrix} \delta_1^0 & & & & & & & & & \\ \delta_1^1 & \delta_1^2 & & & & & & & & \\ \delta_1^2 & \delta_1^3 & \delta_1^4 & & & & & & & \\ \delta_1^3 & \delta_1^4 & \delta_1^5 & \delta_1^6 & \text{Symmetric} & & & & & \\ 0 & 0 & 0 & 0 & \delta_1^0 & & & & & \\ 0 & 0 & 0 & 0 & \delta_1^1 & \delta_1^2 & & & & \\ 0 & 0 & 0 & 0 & \delta_1^2 & \delta_1^3 & \delta_1^4 & & & \\ 0 & 0 & 0 & 0 & \delta_1^3 & \delta_1^4 & \delta_1^5 & \delta_1^6 & & \\ 0 & 0 & 0 & 0 & \delta_1^4 & \delta_1^5 & \delta_1^6 & \delta_1^7 & \delta_1^8 & \\ 0 & 0 & 0 & 0 & \delta_1^5 & \delta_1^6 & \delta_1^7 & \delta_1^8 & \delta_1^9 & \delta_1^{10} \end{bmatrix} \quad (63)$$

10x10

2. Transform the mass matrix to grid point degrees of freedom:

$$[\tilde{M}] = [\bar{\Gamma}_{\beta q}]^T [\tilde{M}] [\bar{\Gamma}_{\beta q}] \quad (64)$$

3. Transform the mass matrix from local to system coordinates:

$$[M] = [\Gamma_{rs}]^T [\tilde{M}] [\Gamma_{rs}] \quad (65)$$

4. The mass matrix,  $[M]$ , is partitioned into 6x6 matrices, and these 6x6 partitions are transformed to local coordinates (see Equations 61 and 62).

# MODULE FUNCTIONAL DESCRIPTIONS

## 4.87.12.7 Thermal Load Calculations (Subroutine TTØRDR of Module SSG1)

1. Compute the temperature gradient constants:

$$\Delta T_1^{(m)} = T_1^{(m)} - T_0, \quad (66)$$

$$\Delta T_2^{(m)} = T_2^{(m)} - T_1^{(m)}, \quad (67)$$

$$\Delta T_1^{(f)} = 0, \quad (68)$$

$$\Delta T_2^{(f)} = 0, \quad (69)$$

where  $T_i^{(m)}$  are the membrane temperatures at point  $i$ .

2. Compute the thermal strain vectors:

$$\{\epsilon_T^{(0)}\} = \Delta T_1^{(m)} \{ALF\}, \quad (70)$$

$$\{\epsilon_T^{(1)}\} = \Delta T_2^{(m)} \{ALF\}, \quad (71)$$

$$\{H_T^{(0)}\} = \Delta T_1^{(f)} \{ALF\}, \quad (72)$$

$$\{H_T^{(1)}\} = \Delta T_2^{(f)} \{ALF\}, \quad (73)$$

where  $\{ALF\}$  is a vector of length two; the first component is  $\alpha_r$  and the second  $\alpha_\theta$ .

3. Form the matrices of integrals

$$[\tilde{F}_{ME}^{(0)}] = 2\pi H_m \begin{bmatrix} 0 & \delta_1^0 & 2\delta_1^1 & 3\delta_1^2 & \lambda_1 \delta_1^0 & \lambda_1 \delta_1^1 & \lambda_1 \delta_1^2 & \lambda_1 \delta_1^3 & \lambda_1 \delta_1^4 & \lambda_1 \delta_1^5 \\ \delta_4^0 & \delta_4^1 & \delta_4^2 & \delta_4^3 & \delta_2^0 & \delta_2^1 & \delta_2^2 & \delta_2^3 & \delta_2^4 & \delta_2^5 \end{bmatrix}_{2 \times 10} \quad (74)$$



# STRUCTURAL ELEMENT DESCRIPTIONS

$$[\tilde{F}_{ME}^{(1)}] = \frac{2\pi H_m}{S} \begin{bmatrix} 0 & \delta_1^1 & 2\delta_1^2 & 3\delta_1^3 & \lambda_1 \delta_1^1 & \lambda_1 \delta_1^2 & \lambda_1 \delta_1^3 & \lambda_1 \delta_1^4 & \lambda_1 \delta_1^5 & \lambda_1 \delta_1^6 \\ \delta_4^1 & \delta_4^2 & \delta_4^3 & \delta_4^4 & \delta_2^1 & \delta_2^2 & \delta_2^3 & \delta_2^4 & \delta_2^5 & \delta_2^6 \end{bmatrix} \quad (75)$$

$$[\tilde{F}_{FE}^{(0)}] = \frac{2\pi H_f^3}{12} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & -2\delta_1^0 & -6\delta_1^1 & -12\delta_1^2 & -20\delta_1^3 \\ 0 & 0 & 0 & 0 & 0 & -\delta_4^0 & -2\delta_4^1 & -3\delta_4^2 & -4\delta_4^3 & -5\delta_4^4 \end{bmatrix} \quad (76)$$

$$[\tilde{F}_{FE}^{(1)}] = \frac{2\pi H_f^3}{12s} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & -2\delta_1^1 & -6\delta_1^2 & -12\delta_1^3 & -20\delta_1^4 \\ 0 & 0 & 0 & 0 & 0 & -\delta_4^1 & -2\delta_4^2 & -3\delta_4^3 & -4\delta_4^4 & -5\delta_4^5 \end{bmatrix} \quad (77)$$

4. Compute the thermal load vector in field coordinates

$$\begin{aligned} \{\tilde{F}_T\} &= [\tilde{F}_{ME}^{(0)}]^T [E] \{\epsilon_T^{(0)}\} + [\tilde{F}_{ME}^{(1)}]^T [E] \{\epsilon_T^{(1)}\} \\ &+ [\tilde{F}_{FE}^{(0)}]^T [E] \{H_T^{(0)}\} + [\tilde{F}_{FE}^{(1)}]^T [E] \{H_T^{(0)}\} . \end{aligned} \quad (78)$$

5. Transform the thermal load vector to grid point degrees of freedom:

$$\{\bar{F}_T\} = [\Gamma_{\beta q}]^T \{\tilde{F}_T\} . \quad (79)$$

6. Transform the thermal load vector from local to system coordinates:

$$\{F_T\} = [\Gamma_{rs}]^T \{\bar{F}_T\} . \quad (80)$$

7. Transform the thermal load vector to basic coordinates:

$$\{F_T\}_b = [T_{12}]^T \{F_T\} , \quad (81)$$

# MODULE FUNCTIONAL DESCRIPTIONS

where

$$[T_{12}] = \begin{bmatrix} T_1 & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & T_2 & 0 \\ 0 & 0 & 0 & I \end{bmatrix}, \quad (82)$$

and  $[T_i]$  and  $[I]$  are 3 by 3 matrices.

8. These vectors are added to the overall load vector,  $\{P_g\}$ .

## 4.87.12.8 Element Force and Stress Calculations (Subroutines STØRD1 and STØRD2 of Module SDR1)

Element stress and force data are calculated in two phases. The first phase (subroutine STØRD1) calculates the element stiffness and stress matrices. The second phase (subroutine STØRD2) calculates the element forces and stresses from the various subcase displacement vectors. Stresses are evaluated at both ends and at the mid-span of the element.

Phase 1 calculations are as follows:

1. Form the element stiffness matrix,  $[K]$ , as in section 4.87.12.5.
2. Set up the coordinates of the three stress points:

$$\{X\} = \begin{pmatrix} 0 \\ \frac{S}{2} \\ S \end{pmatrix}. \quad (83)$$

3. Compute the constants  $\lambda_2, \lambda_3, \lambda_4$  as a function of the stress points coordinates.
  - a. If the element is a toroidal ring, then:

$$\lambda_2^{(i)} = \frac{\cos \left( \alpha_1 + \frac{X(i)}{R_p} \right)}{R_1 - R_p \left[ \sin \alpha_1 - \sin \left( \alpha_1 + \frac{X(i)}{R_p} \right) \right]}, \quad (84)$$

# STRUCTURAL ELEMENT DESCRIPTIONS

$$\lambda_3^{(i)} = \frac{\sin \left( \alpha_1 + \frac{X(i)}{R_p} \right)}{R_1 - R_p \left[ \sin \alpha_1 - \sin \left( \alpha_1 + \frac{X(i)}{R_p} \right) \right]} \quad i = 1, 2, 3, \quad (85)$$

$$\lambda_4^{(i)} = -\lambda_3^{(i)} / R_p . \quad (86)$$

b. If the element is a cylindrical or conical ring, then:

$$\lambda_2^{(i)} = \frac{\cos \alpha_1}{R_1 + X(i) \cos \alpha_1} , \quad (87)$$

$$\lambda_3^{(i)} = \frac{\cos \alpha_1}{R_1 + X(i) \cos \alpha_1} \quad i = 1, 2, 3, \quad (88)$$

$$\lambda_4^{(i)} = 0 . \quad (89)$$

c. If the element is a shell cap, then:

$$\lambda_2^{(i)} = \frac{\cos \left( \alpha_1 + \frac{X(i)}{R_p} \right)}{R_1 - R_p \left[ \sin \alpha_1 - \sin \left( \alpha_1 + \frac{X(i)}{R_p} \right) \right]} \quad i = 1, 2, 3, \quad (90)$$

$$\lambda_3^{(i)} = 1/R_p , \quad (91)$$

$$\lambda_4^{(i)} = -1/(R_p^2) . \quad (92)$$

4. Compute the stress matrix in field coordinates for the three stress points as shown in Tables 6 and 7. Note that the factors  $H$  and  $H^3/12$  have been omitted for  $[\tilde{S}_1^{(i)}]$  and  $[\tilde{S}_2^{(i)}]$  respectively.

If the element is a shell cap, then modify  $[\tilde{S}^{(i)}]$  by:

$$\tilde{S}_{12} = E_{12} - E_{11} , \quad (93)$$



# MODULE FUNCTIONAL DESCRIPTIONS

$$\tilde{S}_{22} = E_{22} + E_{12} , \quad (94)$$

$$\tilde{S}_{37} = -2 (E_{12} + E_{11}) , \quad (95)$$

$$\tilde{S}_{47} = 2 (E_{22} + E_{12}) , \quad (96)$$

$$\tilde{S}_{58} = 3(E_{22} - 4 E_{11}) . \quad (97)$$

5. Form the master stress matrix in field coordinates:

$$[\tilde{S}] = \begin{bmatrix} \tilde{S}^{(1)} \\ \tilde{S}^{(2)} \\ \tilde{S}^{(3)} \end{bmatrix} \quad 15 \times 10 . \quad (98)$$

# STRUCTURAL ELEMENT DESCRIPTIONS

Table 6. Terms in Columns 1 Through 6 of the 2 by 10  $[\tilde{S}_1^{(i)}]$  Matrix and the 3 by 10  $[\tilde{S}_2^{(i)}]$  Matrix.

The  $[\tilde{S}_1^{(i)}]$  Matrix

| Col. 1                   | Col. 2                         | Col. 3                           | Col. 4                           | Col. 5                     | Col. 6                         |
|--------------------------|--------------------------------|----------------------------------|----------------------------------|----------------------------|--------------------------------|
| $\lambda_2^{(i)} E_{12}$ | $E_{11}$                       | $2E_{11}X(i)$                    | $3E_{11}X^2(i)$                  | $\lambda_1^{(i)} E_{11}$   | $(\lambda_1^{(i)} E_{11})$     |
|                          | $+ \lambda_2^{(i)} E_{12}X(i)$ | $+ \lambda_2^{(i)} E_{12}X^2(i)$ | $+ \lambda_2^{(i)} E_{12}X^3(i)$ | $+ \lambda_3^{(i)} E_{12}$ | $+ \lambda_3^{(i)} E_{12}X(i)$ |
| $\lambda_2^{(i)} E_{22}$ | $E_{12}$                       | $2E_{12}X^2(i)$                  | $3E_{12}X^2(i)$                  | $\lambda_1^{(i)} E_{12}$   | $(\lambda_1^{(i)} E_{12})$     |
|                          | $+ \lambda_2^{(i)} E_{22}X(i)$ | $+ \lambda_2^{(i)} E_{22}X^2(i)$ | $+ \lambda_2^{(i)} E_{22}X^3(i)$ | $+ \lambda_3^{(i)} E_{22}$ | $+ \lambda_3^{(i)} E_{22}X(i)$ |

The  $[\tilde{S}_2^{(i)}]$  Matrix

|   |   |   |   |   |                            |
|---|---|---|---|---|----------------------------|
| 0 | 0 | 0 | 0 | 0 | $- \lambda_2^{(i)} E_{12}$ |
| 0 | 0 | 0 | 0 | 0 | $\lambda_2^{(i)} E_{22}$   |
| 0 | 0 | 0 | 0 | 0 | $g(i)$                     |

where  $g(i) = (\lambda_2^{(i)})^2 E_{22} - \lambda_4^{(i)} E_{12}$  and

$$[\tilde{S}^{(i)}] = \begin{bmatrix} \tilde{S}_1^{(i)} \\ \tilde{S}_2^{(i)} \end{bmatrix}_{5 \times 10}$$

# MODULE FUNCTIONAL DESCRIPTIONS

Table 7. Terms in Columns 7 Through 10 of the 2 by 10  $[\tilde{S}_1^{(i)}]$  Matrix and the 3 by 10  $[\tilde{S}_2^{(i)}]$  Matrix.

The  $[\tilde{S}_1^{(i)}]$  Matrix

|       | Col. 1  | Col. 2  | Col. 3  | Col. 4  |
|-------|---|---|---|---|
| Row 1 | $(\lambda_1^{(i)} E_{11})$<br>$+ \lambda_3^{(i)} E_{12} X^2(i)$ | $(\lambda_1^{(i)} E_{11})$<br>$+ \lambda_3^{(i)} E_{12} X^3(i)$ | $(\lambda_1^{(i)} E_{11})$<br>$+ \lambda_3^{(i)} E_{12} X^4(i)$ | $(\lambda_1^{(i)} E_{11})$<br>$+ \lambda_3^{(i)} E_{12} X^5(i)$ |
| Row 2 | $(\lambda_1^{(i)} E_{12})$<br>$+ \lambda_3^{(i)} E_{22} X^2(i)$ | $(\lambda_1^{(i)} E_{12})$<br>$+ \lambda_3^{(i)} E_{22} X^3(i)$ | $(\lambda_1^{(i)} E_{12})$<br>$+ \lambda_3^{(i)} E_{22} X^4(i)$ | $(\lambda_1^{(i)} E_{12})$<br>$+ \lambda_3^{(i)} E_{22} X^5(i)$ |

The  $[\tilde{S}_2^{(i)}]$  Matrix

|       |  |  |   |   |
|-------|--|--|---|---|
| Row 1 | $-2\lambda_2^{(i)} E_{12} X(i)$<br>$- 2E_{11}$ | $- 3\lambda_2^{(i)} E_{12} X^2(i)$<br>$- 6E_{11} X(i)$               | $- 4\lambda_2^{(i)} E_{12} X^3(i)$<br>$- 12E_{11} X^2(i)$                     | $- 5\lambda_2^{(i)} E_{12} X^4(i)$<br>$- 20E_{11} X^3(i)$                       |
| Row 2 | $2\lambda_2^{(i)} E_{22} X(i)$<br>$+ 2E_{12}$  | $3\lambda_2^{(i)} E_{12} X^2(i)$<br>$+ 6E_{12} X(i)$                 | $4\lambda_2^{(i)} E_{22} X^3(i)$<br>$+ 12E_{12} X^2(i)$                       | $5\lambda_2^{(i)} E_{22} X^4(i)$<br>$+ 20E_{12} X^3(i)$                         |
| Row 3 | $2g^{(i)} X(i)$<br>$- 2\lambda_2^{(i)} E_{11}$ | $3g^{(i)} X^2(i)$<br>$- 6\lambda_2^{(i)} E_{11} X(i)$<br>$- 6E_{11}$ | $4g^{(i)} X^3(i)$<br>$- 12\lambda_2^{(i)} E_{11} X^2(i)$<br>$- 24E_{11} X(i)$ | $5g^{(i)} X^4(i)$<br>$- 20\lambda_2^{(i)} E_{11} X^3(i)$<br>$- 60E_{11} X^2(i)$ |

where

$$g^{(i)} = (\lambda_2^{(i)})^2 E_{22} - \lambda_4^{(i)} E_{12}$$

and

$$[\tilde{S}^{(i)}] = \begin{bmatrix} \tilde{S}_1^{(i)} \\ \tilde{S}_2^{(i)} \end{bmatrix}$$



# STRUCTURAL ELEMENT DESCRIPTIONS

6. Transform the stress matrix to grid point degrees of freedom

$$[\bar{S}] = [\tilde{S}][\bar{\Gamma}_{\beta q}] . \quad (99)$$

7. Transform the stress matrix from local to system coordinates

$$[S] = [\bar{S}] [\Gamma_{rs}] . \quad (100)$$

8. Transform the stress matrix to global coordinates

$$[S]_g = [S] [T_{12}] . \quad (101)$$

9. Compute the thermal stress vector for the three stress points:

$$\{T_s^{(i)}\} = \left\{ \begin{array}{l} \Delta T_1^{(m)} H_m (E_{11}\alpha_1 + E_{12}\alpha_2) + \Delta T_2^{(m)} H_m \frac{\chi_i}{S} (E_{11}\alpha_1 + E_{12}\alpha_2) \\ \Delta T_1^{(m)} H_m (E_{21}\alpha_1 + E_{22}\alpha_2) + \Delta T_2^{(m)} H_m \frac{\chi_i}{S} (E_{21}\alpha_1 + E_{22}\alpha_2) \\ \Delta T_1^{(f)} \frac{H_f^3}{12} (E_{11}\alpha_1 + E_{12}\alpha_2) + \Delta T_2^{(f)} \frac{H_f^3 \chi_i}{12S} (E_{11}\alpha_1 + E_{12}\alpha_2) \\ -\Delta T_1^{(f)} \frac{H_f^3}{12} (E_{21}\alpha_1 + E_{22}\alpha_2) - \Delta T_2^{(f)} \frac{H_f^3 \chi_i}{12S} (E_{21}\alpha_1 + E_{22}\alpha_2) \\ \Delta T_1^{(f)} \frac{H_f^3}{12} \lambda_2^{(i)} [(E_{11} - E_{12})\alpha_1 + (E_{12} - E_{22})\alpha_2] \\ + \Delta T_2^{(f)} \frac{H_f^3}{12S} \left\{ \lambda_2^{(i)} \chi_i [(E_{11} - E_{12})\alpha_1 \right. \\ \left. + (E_{12} - E_{22})\alpha_2] + [E_{11}\alpha_1 + E_{12}\alpha_2] \right\} \end{array} \right\} \quad \left. \begin{array}{l} (5 \times 1) \\ (102) \end{array} \right\}$$

where  $\Delta T_1^{(m)}$ ,  $\Delta T_2^{(m)}$ ,  $\Delta T_1^{(f)}$  and  $\Delta T_2^{(f)}$  are as given in Equations 66 through 69.

# MODULE FUNCTIONAL DESCRIPTIONS

10. Form the master thermal stress vector

$$\{T_s\} = \begin{pmatrix} T_s^{(1)} \\ T_s^{(2)} \\ T_s^{(3)} \end{pmatrix} \cdot \quad (103)$$

Phase 2 calculations are as follows:

1. Extract the displacement vector,  $\{\Delta\}$ , at the two grid points from the global displacement vector.

2. Calculate the element forces:

$$\{P\} = [K]\{\Delta\} \cdot \quad (104)$$

3. Calculate the element stresses without regard to thermal loading:

$$\{\sigma'\} = [S]_g \{\Delta\} \quad (12 \times 1) \cdot \quad (105)$$

4. If there is no thermal loading, then

$$\{\sigma\} = \{\sigma'\} \cdot \quad (106)$$

5. If there is thermal loading, then

$$\sigma_j = \sigma_j^i - (T_1 - T_0) T_{Sj} - (T_2 - T_1) T_{Sj + 15} \cdot \quad (107)$$

for  $j = 1$  and  $2$  and

$$\sigma_j = \sigma_j^i \cdot \quad (108)$$

for  $j = 3, 4, 5$ .

## STRUCTURAL ELEMENT DESCRIPTIONS

### 4.87.13 The VISC Element

The viscous element, VISC, has the same properties as the RØD element (see section 4.87.1) except that instead of generating a contribution to the stiffness matrix, the element generates a contribution to the damping matrix,  $[B_{gg}]$ .

#### 4.87.13.1 Input Data for the VISC Element

1. The ECPT/EST entries for the VISC are:

| <u>Symbol</u>   | <u>Description</u>   |
|---|--|
| $SIL_a, SIL_b$  | Scalar indices for grid points a and b                               |
| $\left. \begin{array}{l} N_a, X_a, Y_a, Z_a \\ N_b, X_b, Y_b, Z_b \end{array} \right\}$ | Local coordinate system number and basic coordinates of grid points. |
| $C_1$   | Extensional damping coefficient                                      |
| $C_2$   | Torsional damping coefficient  |

Given  $N_a, X_a, Y_a, Z_a, N_b, X_b, Y_b$  and  $Z_b$  and the CSTM (Coordinate System Transformation Matrices) data block, the 3 by 3 transformation matrices,  $[T_a]$  and  $[T_b]$ , are calculated using utility routine TRANSF (see section 3.4.37).

#### 4.87.13.2 Damping Matrix Calculations (Subroutine BVISC of Module SMA2)

1. Generate  $\{n\}$  as in Equation 2, section 4.87.1.2, and generate the transformation matrices  $[T_a]$  and  $[T_b]$ .
2. Calculate:

$$[b_t] = C_1 \begin{bmatrix} n_1^2 & n_1 n_2 & n_1 n_3 \\ n_1 n_2 & n_2^2 & n_2 n_3 \\ n_1 n_3 & n_2 n_3 & n_3^2 \end{bmatrix}, \quad (1)$$



# MODULE FUNCTIONAL DESCRIPTIONS

$$[b_r] = c_2 \begin{bmatrix} n_1^2 & n_1 n_2 & n_1 n_3 \\ n_1 n_2 & n_2^2 & n_2 n_3 \\ n_1 n_3 & n_2 n_3 & n_3^2 \end{bmatrix} . \quad (2)$$

3. The 6x6 damping matrices are:

$$[B_{aa}] = \begin{bmatrix} T_a^T b_t T_a & 0 \\ 0 & T_a^T b_r T_a \end{bmatrix} , \quad (3)$$

$$[B_{ab}] = - \begin{bmatrix} T_a^T b_t T_b & 0 \\ 0 & T_a^T b_r T_b \end{bmatrix} , \quad (4)$$

$$[B_{bb}] = \begin{bmatrix} T_b^T b_t T_b & 0 \\ 0 & T_b^T b_r T_b \end{bmatrix} , \quad (5)$$

$$[B_{ba}] = - \begin{bmatrix} T_b^T b_t T_a & 0 \\ 0 & T_b^T b_r T_a \end{bmatrix} . \quad (6)$$

4. These matrices are added to  $[B_{gg}]$ .

## STRUCTURAL ELEMENT DESCRIPTIONS

### 4.87.14 Integral Calculations for the TRIARG, TRAPRG Elements

Integrals of the form

$$I_{pq} = \int_{R_j}^{R_i} \int_{Z_{mn}}^{Z_{k\ell}} r^p z^q dz dr \quad . \quad (1)$$

must be calculated for the TRIARG and TRAPRG elements (see sections 4.87.10.3 and 4.87.11.3). The integration may be performed for any integer values of  $p$  and  $q$ . The area of integration is defined by the two lines  $r = R_i$  and  $r = R_j$ , and by the two lines  $z = b_{k\ell}r + a_{k\ell}$  and  $z = b_{mn}r + a_{mn}$ .

This integration is performed by the integration "driver" subroutine, FØRTRAN function DK1 in module SMA1, FØRTRAN function DMI in module SMA2, and FØRTRAN function AI in module SDR2.

The following input data are necessary for these routines:

- $p$  - an integer that defines the power of the  $r$  variable.
- $q$  - an integer that defines the power of the  $z$  variable.
- $\{R\}$  - a vector of the  $r$  coordinates of all points used to describe the area of integration.
- $\{Z\}$  - a vector of the  $z$  coordinates of all points used to describe the area of integration.
- $k, \ell$  - the subscripts of  $R, Z$  defining one of the lines of the limit of integration (i.e., the line between points  $(r_k, z_k)$  and  $(r_\ell, z_\ell)$ ).
- $m, n$  - the subscripts of  $R, Z$  defining the second line on the limit of integration.
- $i, j$  - the subscripts of  $R$  defining the other two lines on the limit of integration.

In the following paragraphs FØRTRAN names of functions auxiliary to DK1 are given. The corresponding FØRTRAN function names auxiliary to functions DMI and AI can be found in sections 4.28.8 and 4.46.8 respectively.

# MODULE FUNCTIONAL DESCRIPTIONS

The following slopes and y-intercepts are calculated in functions DKK and DKM

$$a_{k\ell} = \frac{R_{\ell}Z_k - R_kZ_{\ell}}{R_{\ell} - R_k}, \quad (2)$$

$$b_{k\ell} = \frac{Z_{\ell} - Z_k}{R_{\ell} - R_k}, \quad (3)$$

$$a_{mn} = \frac{R_nZ_m - R_mZ_n}{R_n - R_m}, \quad (4)$$

$$b_{mn} = \frac{Z_n - Z_m}{R_n - R_m}. \quad (5)$$

A test for a vanishing area of integration is made: if  $R_i = R_j$ , then  $I_{pq} = 0$ ;  
if  $a_{k\ell} = a_{mn}$  and  $b_{k\ell} = b_{mn}$ , then  $I_{pq} = 0$ .

The formulas for evaluation of the integrals are dependent upon the values of p and q as given in the following sections.



# STRUCTURAL ELEMENT DESCRIPTIONS

## 4.87.14.1 Integral Calculation for $q \geq 0$ and any $p$ . (Function DKINT)

Define the function

$$f_1(x, y) = \frac{1}{q+1} \sum_{t=0}^{q+1} C x^t y^{q+1-t} D, \quad (6)$$

where

$$C = \begin{cases} \prod_{s=1}^t \frac{q+1-s+1}{s} & \text{for } t \neq 0, \\ 1 & \text{for } t = 0, \end{cases} \quad (7)$$

$$D = \begin{cases} \left[ \frac{R_j^{(q+1+p+1-t)} - R_i^{(q+1+p+1-t)}}{q+1+p+1-t} \right] & \text{for } (q+1+p+1-t) \neq 0 \\ \ln(R_j/R_i) & \text{for } (q+1+p+1-t) = 0 \end{cases} \quad (8)$$

C and D are calculated in functions DKEF and DKJ respectively.

The integral is

$$I_{pq} = f_1(a_{mn}, b_{mn}) - f_1(a_{kl}, b_{kl}). \quad (9)$$

## 4.87.14.2 Integral Calculation for $p \geq 0$ and $q < -1$ (Function DK89)

$$f_2(\alpha, x, y) = \frac{1}{y^{p+1}} \sum_{s=0}^p p! (-x)^s D, \quad (10)$$

where

$$D = \begin{cases} \frac{(x+yR_\alpha)^{p+1+q+1-s}}{(p-s)!s!(p+1+q+1-s)} & \text{for } (p+1+q+1-s) \neq 0 \\ \frac{\ln|x+yR_\alpha|}{(p+1+q+1)!(-q-2)!} & \text{for } (p+1+q+1-s) = 0 \end{cases}, \quad (11)$$

and  $\alpha = i$  or  $j$ .

The integral is

$$I_{pq} = \frac{1}{2+q} [f_2(i, a_{k\ell}, b_{k\ell}) - f_2(i, a_{mn}, b_{mn}) - f_2(j, a_{k\ell}, b_{k\ell}) + f_2(j, a_{mn}, b_{mn})] \quad (12)$$

#### 4.87.14.3 Integral Calculation for $p < 0$ and $q < -1$ (Function DK100)

Define the function

$$f_3(\alpha, x, y) = \frac{-1}{x^{(-p-q-2)}} \sum_{s=0}^{-p-q-3} (-p-q-3)! D, \quad (13)$$

where

$$D = \begin{cases} \frac{(x+yR_\alpha)^{(-p-1-s)} (-y)^s}{(-p-q-3-s)! s! (-p-1-s) R_\alpha^{(-p-1-s)}} & \text{for } (-p-1-s) \neq 0 \\ \frac{(-y)^{-p-1} \ln \left( \left| \frac{x+yR_\alpha}{R_\alpha} \right| \right)}{(-p-1)! (-q-2)!} & \text{for } (-p-1-s) = 0 \end{cases}, \quad (14)$$

and  $\alpha = i$  or  $j$ .

The integral is

$$I_{pq} = \frac{1}{2+q} [f_3(i, a_{k\ell}, b_{k\ell}) - f_3(i, a_{mn}, b_{mn}) - f_3(j, a_{k\ell}, b_{k\ell}) + f_3(j, a_{mn}, b_{mn})] \quad (15)$$

# STRUCTURAL ELEMENT DESCRIPTIONS

## 4.87.14.4 Integral Calculations for $p > -1$ and $q = -1$ (Function DKJAB)

Define the function

$$f_4(\alpha, x, y) = \frac{R_\alpha^{p+1} \ln(|x+yR_\alpha|)}{p+1} - \frac{y}{p+1} f_2(\alpha, x, y), \quad (16)$$

where  $f_2$  is given in Equation 10.

The integral is

$$\begin{aligned} I_{pq} = & f_4(i, a_{kl}, b_{kl}) - f_4(i, a_{mn}, b_{mn}) \\ & - f_4(j, a_{kl}, b_{kl}) - f_4(j, a_{mn}, b_{mn}) \end{aligned} \quad (17)$$

## 4.87.14.5 Integral Calculations for $p < -1$ and $q = -1$ (Function DK219)

Define the function

$$f_5(\alpha, x, y) = -\frac{\ln(|x+yR_\alpha|)}{(-p-1)R_\alpha^{(-p-1)}} + \frac{y}{(-p-1)} f_3(\alpha, x, y), \quad (18)$$

where  $f_3$  is given in Equation 13.

The integral is

$$\begin{aligned} I_{pq} = & f_5(i, a_{kl}, b_{kl}) - f_5(i, a_{mn}, b_{mn}) \\ & - f_5(j, a_{kl}, b_{kl}) - f_5(j, a_{mn}, b_{mn}) \end{aligned} \quad (19)$$



# MODULE FUNCTIONAL DESCRIPTIONS

## 4.87.14.6 Integral Calculations for $p = -1$ and $q = -1$ (Function DK211)

Define the function

$$f_6(\alpha, x, y) = \begin{cases} 0, & \text{for } yR_\alpha = x \\ \frac{1}{2} [\ln(|2 y R_\alpha|)]^2, & \text{for } (yR_\alpha)^2 = x^2 \\ & \text{and } yR_\alpha \neq x \\ \ln|x| \ln|R_\alpha| - \sum_{t=1}^{\infty} \frac{1}{t^2} \left[ \frac{-yR_\alpha}{x} \right]^t, & \text{for } (yR_\alpha)^2 < x^2 \\ \frac{1}{2} [\ln(|yR_\alpha|)]^2 + \sum_{t=1}^{\infty} \frac{1}{t^2} \left[ \frac{-x}{yR_\alpha} \right]^t, & \text{for } (yR_\alpha)^2 < x^2 \end{cases} \quad (20)$$

The summation term is calculated until its value becomes less than  $1.0 \times 10^{-6}$ .

The integral is

$$\begin{aligned} I_{pq} &= f_6(i, a_{kl}, b_{kl}) - f_6(i, a_{mn}, b_{mn}) \\ &\quad - f_6(i, a_{kl}, b_{kl}) - f_6(i, a_{mn}, b_{mn}) \end{aligned} \quad (21)$$

## STRUCTURAL ELEMENT DESCRIPTIONS

### 4.87.15 The FLUID2, FLUID3, FLUID4, AXIF2, AXIF3, AXIF4, and MFREE Elements

#### 4.87.15.1 Input Data for the Fluid Elements

1. The ECPT/EST entries for the FLUID2 element are:

| <u>Symbol</u>                        | <u>Description</u>  |
|--------------------------------------|---|
| $SIL_1, SIL_2$                       | Scalar indices for the connected scalar points  |
| $N_i = 0, r_i, z_i, 0$<br>$i = 1, 2$ | Reference number for the basic coordinate system and locations in the fluid coordinate systems. |
|                                      |   |
| $\rho$                               | Fluid density   |
| $B$                                  | Fluid bulk modulus  |
| $n$                                  | Harmonic number   |

2. The ECPT/EST entries for the FLUID3 and FLUID4 elements are identical except that three and four points are used respectively.
3. The ECPT/EST entries for the MFREE element are identical to the FLUID2 element except that a weight factor,  $\gamma$ , is used instead of  $\rho$  and  $B$ .
4. No other material or coordinate system data is necessary.
5. The AXIF elements are identical to the FLUID elements at this stage.

#### 4.87.15.2 Matrix Calculations for the FLUID2 Element (Subroutine KFLUD2 of Module SMA1 and Subroutine MFLUD2 of Module SMA2)

The FLUID2 element is intended to model a fluid in the region adjacent to and including the axis of symmetry. The volume is defined by two circular ring points in the fluid. The shape is that of a disc having a conical or cylindrical outer boundary.

1. The integral parameters, for the stiffness matrix,  $I_{2n,0}$ ,  $I_{2n,1}$ ,  $I_{2n,2}$ , and  $I_{2n+2,0}$  are calculated according to the following equations:

$$\left. \begin{aligned} I_{2n,0} &= I_{2n,1} = I_{2n,2} = 0, \\ I_{2n+2,0} &= \frac{1}{6} (z_2 - z_1)(r_2^2 + r_1 r_2 + r_1^2) \end{aligned} \right\} n = 0. \quad (1)$$

MODULE FUNCTIONAL DESCRIPTIONS

$$\text{If } \left| \frac{r_2 - r_1}{z_2 - z_1} \right| < 10^{-6} :$$

$$\left. \begin{aligned} I_{2n,0} &= \left[ \frac{r_1 + r_2}{2} \right]^{2n} \left( \frac{z_2 - z_1}{2n} \right) \\ I_{2n,1} &= \frac{I_{2n,0}}{2} (z_2 + z_1) \\ I_{2n,2} &= \frac{I_{2n,0}}{3} (z_2^2 + z_1 z_2 + z_1^2) \\ I_{2n+2,0} &= I_{2n,0} \left( \frac{2n}{2n+2} \right) r_1^2 \end{aligned} \right\} n > 0 . \quad (2)$$

$$\text{If } \left| \frac{r_2 - r_1}{z_2 - z_1} \right| > 10^{-6} :$$

$$\begin{aligned} D &= \frac{z_2 - z_1}{r_2 - r_1} \\ I_{2n,0} &= \frac{D}{2n(2n+1)} (r_2^{2n+1} - r_1^{2n+1}) \\ I_{2n,1} &= \frac{D}{2n(2n+1)} \left[ r_2^{2n+1} z_2 - r_1^{2n+1} z_1 - \left( \frac{D}{2n+2} \right) (r_2^{2n+2} - r_1^{2n+2}) \right] \\ I_{2n,2} &= \frac{D}{2n(2n+1)} \left\{ r_2^{2n+1} z_2^2 - r_1^{2n+1} z_1^2 - \left( \frac{2D}{2n+2} \right) [r_2^{2n+2} z_2 - r_1^{2n+2} z_1] \right. \\ &\quad \left. - \frac{D}{2n+3} (r_2^{2n+3} - r_1^{2n+3}) \right\} \\ I_{2n+2,0} &= \frac{D}{(2n+2)(2n+3)} [r_2^{2n+3} - r_1^{2n+3}] \end{aligned} \quad \left. \begin{aligned} & \\ & \\ & \\ & \\ & \end{aligned} \right\} n > 0 \quad (4)$$

2. The integral parameters for the mass matrix,  $I_{2n+2,0}$ ,  $I_{2n+2,1}$ , and  $I_{2n+2,2}$  are calculated with the same equations as above except the value  $k = 2n+2$  is substituted for  $k = 2n$ .
3. The transformation matrix  $[H_{qp}^n]$  is defined as:



# STRUCTURAL ELEMENT DESCRIPTIONS

$$[H_{pq}^n] = \frac{1}{z_2 - z_1} \begin{bmatrix} \frac{z_2}{r_1^n} & -\frac{z_1}{r_2^n} \\ -\frac{1}{r_1^n} & \frac{1}{r_2^n} \end{bmatrix} \quad (5)$$

4. The stiffness matrix is:

$$[K_p^n] = \frac{\pi}{\rho} [H_{pq}^n]^T \begin{bmatrix} (2n^2 I_{2n,0}) & (2n^2 I_{2n,1}) \\ (2n^2 I_{2n,1}) & (2n^2 I_{2n,2} + I_{2n+2,0}) \end{bmatrix} [H_{pq}^n] \quad (6)$$

$n > 0$

Note: if  $n = 0$ , a factor of 2 is used.

5. The mass matrix is:

$$[M_p^n] = \frac{\pi}{B} [H_{pq}^n]^T \begin{bmatrix} I_{2n+2,0} & I_{2n+2,1} \\ I_{2n+2,1} & I_{2n+2,2} \end{bmatrix} [H_{pq}^n], \quad n > 0 \quad (7)$$

Note: if  $n = 0$  a factor of 2 is used.

6. Various tests are performed for the element.

If  $|z_2 - z_1| = 0$ , the calculations are skipped,

if  $r_1 = 0$  or  $r_2 = 0$ , a fatal error exists,

if  $\rho = 0$ , a fatal error exists,

if  $B = 0$ , the mass calculations are skipped.

# MODULE FUNCTIONAL DESCRIPTIONS

## 4.87.15.3 Matrix Calculations for the FLUID3 Element (Subroutines KFLUD3 of Module SMA1 and Subroutine MFLUD3 of Module SMA2)

The FLUID3 element is used to model a volume of fluid defined by three connected fluid ring points.

1. The three connected points are arranged in the order such that the area factor,  $R$ , is positive. The area factor is defined by the equation:

$$R = (r_2 - r_1)(z_3 - z_1) - (r_3 - r_1)(z_2 - z_1) . \quad (8)$$

2. The transformation matrix,  $[H_{pq}]$ , is calculated as:

$$[H_{pq}] = \frac{1}{R} \begin{bmatrix} (r_2 z_3 - r_3 z_2) & (r_3 z_1 - r_1 z_3) & (r_1 z_2 - r_2 z_1) \\ (z_2 - z_3) & (z_3 - z_1) & (z_1 - z_2) \\ (r_3 - r_2) & (r_1 - r_3) & (r_2 - r_1) \end{bmatrix} . \quad (9)$$

3. The integral parameters,  $I_{kl}$ , for the stiffness matrix are the sum of the integrals,  $G_{kl}$ , for each of the three sides. The points defining each side are:

| <u>SIDE</u> | <u>POINTS - a,b</u> |
|-------------|---------------------|
| 1           | 1, 2                |
| 2           | 2, 3                |
| 3           | 3, 1                |

The following parameters are used to generate the integrals,  $G_{kl}$ :

$$\left. \begin{aligned} \Delta r &= r_b - r_a \\ \Delta z &= z_b - z_a \\ \beta &= z_a - r_a \frac{\Delta z}{\Delta r} \end{aligned} \right\} . \quad (10)$$

# STRUCTURAL ELEMENT DESCRIPTIONS

The integrals for each side are:

$$\begin{aligned}
 G_{00} &= \beta \log \frac{r_a}{r_b} - \Delta z \\
 G_{10} &= -\beta \Delta r + \frac{\Delta z}{2\Delta r} (r_a^2 - r_b^2) \\
 G_{20} &= \frac{1}{2} \beta (r_a^2 - r_b^2) + \frac{\Delta z}{3\Delta r} (r_a^3 - r_b^3) \\
 G_{01} &= \frac{1}{2} \beta^2 \log \frac{r_a}{r_b} - \beta \Delta z + \frac{1}{4} \frac{\Delta z^2}{\Delta r^2} (r_a^2 - r_b^2) \\
 G_{11} &= -\frac{1}{2} \beta^2 \Delta r + \frac{1}{2} \beta \frac{\Delta z}{\Delta r} (r_a^2 - r_b^2) + \frac{1}{6} \left( \frac{\Delta z}{\Delta r} \right)^2 (r_a^3 - r_b^3) \\
 G_{02} &= \frac{1}{3} \beta^3 \log \frac{r_a}{r_b} - \beta^2 \Delta z + \frac{1}{2} \beta \left( \frac{\Delta z}{\Delta r} \right)^2 (r_a^2 - r_b^2) + \frac{1}{9} \left( \frac{\Delta z}{\Delta r} \right)^3 (r_a^3 - r_b^3)
 \end{aligned} \tag{11}$$

4. The stiffness matrix is:

$$[K_p^n] = \frac{\pi}{\rho} [H_{pq}]^T \begin{bmatrix} n^2 I_{00} & n^2 I_{10} & n^2 I_{01} \\ n^2 I_{10} & (n^2+1) I_{20} & n^2 I_{11} \\ n^2 I_{01} & n^2 I_{11} & (n^2 I_{02} + I_{20}) \end{bmatrix} [H_{pq}] \tag{12}$$

The matrix terms are multiplied by two if  $n = 0$ .

5. The mass matrix terms are simply:

$$M_{ij}^n = \frac{\pi A}{60B} (r_1 + r_2 + r_3 + r_i + r_j) c_{ij} \tag{13}$$

where

$$\begin{aligned}
 c_{ij} &= 2, & i &= j, \\
 c_{ij} &= 1, & i &\neq j,
 \end{aligned}$$

and

$$A = \frac{R}{2} \text{ is the area.}$$



## MODULE FUNCTIONAL DESCRIPTIONS

6. The tests performed are:

- if  $r_i = 0$  a fatal error exists,
- if  $R = 0$  the routine exits,
- if  $\rho = 0$  a fatal error exists,
- if  $B = 0$  the mass routine exits.

### 4.87.15.4 Matrix Generation for the FLUID4 Element (Subroutine KFLUD4 in Module SMA1 and Subroutine MFLUD4 in Module SMA2)

This element describes an axisymmetric volume of fluid defined by four fluid ring points. It is actually solved by subdividing the quadrilateral cross section into four triangles and calling the appropriate FLUID3 subroutine for each of the triangles. The parameters  $\rho$  and  $B$  are multiplied by two in order to account for the overlapping volumes and reduce the matrix terms.

1. A test is made in the stiffness routine to check the interior angles which must be less than  $180^\circ$ . For each of the four triangles, the area factor  $K$  is calculated which will be positive if the order of the points is counterclockwise. If  $K$  is negative for one or three out of the four triangles, a fatal error exists.
2. The triangles and their three connected points are:

| <u>Triangle</u> | <u>Connected Points</u> |   |   |
|-----------------|-------------------------|---|---|
|                 | a                       | b | c |
| I               | 1                       | 2 | 3 |
| II              | 1                       | 2 | 4 |
| III             | 1                       | 3 | 4 |
| IV              | 2                       | 3 | 4 |

(14)

The ECPT data is moved to a temporary storage space and the original ECPT is used for the data for each triangle.

3. Since matrix terms are only created if one of the connected points is the "pivot point", a test is made and the FLUID3 subroutine is not called if the "pivot point" is not one of the three points.

## STRUCTURAL ELEMENT DESCRIPTIONS

### 4.87.15.5 Matrix Calculations for the MFREE Element (Subroutine MFREE in Module SMA2)

The data for this element is generated by subroutine IFP4 from the free surface information and is not available as a user-input element. The element describes the effect of gravity on a surface in between two fluid ring points. In a special case, the surface is interior to a circle defined by one fluid ring point.

1. If the two connected points are identical ( $SIL_1 = SIL_2$ ), the special case exists and the equations are:

$$\left. \begin{aligned} M_{ii} &= \frac{\pi r^2}{2\gamma(2n+2)}, & n > 0 \\ M_{ii} &= \frac{\pi r^2}{2\gamma}, & n = 0 \end{aligned} \right\} \quad (15)$$

A factor of two is included in the denominator because the terms will be calculated twice.

2. If the connected points are unique, the equation for the mass matrix is:

$$[M_p^n] = \frac{\pi(r_2 - r_1)}{12\gamma} \begin{bmatrix} 3r_1 + r_2 & r_1 + r_2 \\ r_1 + r_2 & 3r_2 + r_1 \end{bmatrix}, \quad n > 0. \quad (16)$$

The values are multiplied by two for  $n = 0$ .

### 4.87.15.6 Stress Calculations for the AXIF Elements, Phase 1.

The SDR2 calculations for these elements are actually the calculations of the velocity of the fluid passing through a fluid element.

The data placed on the ESTB file are:

1.  $Id_e$  - Element Id
2.  $SIL_1, SIL_2, (SIL_3, SIL_4)$  - Scalar indices of connected points
3.  $[S_v]$  - the velocity-pressure matrix

The  $[S_v]$  matrix for the CAXIF2 element is a four by two matrix given as follows:

# MODULE FUNCTIONAL DESCRIPTIONS

$$[S_v] = \begin{bmatrix} S_c^2 \\ S_e^2 \end{bmatrix} \quad (17)$$

where

$$\begin{pmatrix} v_{rc} \\ v_{zc} \\ v_{se} \\ v_{\phi e} \end{pmatrix} = [S_v] \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} \quad (18)$$

$v_{rc}$  and  $v_{zc}$  are velocities at  $r = 0$ ,  $v_{se}$  and  $v_{\phi e}$  are velocities at the midpoint of the outer edge, along the edge and circumferential.

The two by two matrix  $[S_c]$ , for the center, is:

$$[S_c^2] = \frac{1}{\rho} \begin{bmatrix} 0 & 0 \\ \frac{1}{z_2 - z_1} & \frac{-1}{z_2 - z_1} \end{bmatrix} \quad n = 0 \quad (19)$$

$$[S_c^2] = \frac{1}{\rho} \begin{bmatrix} \frac{-1}{r_1 + r_2} & \frac{-1}{r_1 + r_2} \\ 0 & 0 \end{bmatrix} \quad n = 1 \quad (20)$$

$$[S_c^2] = [0] \quad n > 1 \quad (21)$$

The two by two  $[S_e^2]$  matrix, for the outer edge is:

$$[S_e^2] = -\frac{(\bar{r})^{n-1}}{\rho \ell} \begin{bmatrix} n\Delta r & (n\bar{z}\Delta r + \bar{r}\Delta z) \\ n\ell & n\bar{z}\ell \end{bmatrix} [H_{qp}^n] \quad (22)$$



# STRUCTURAL ELEMENT DESCRIPTIONS

where

$$\left. \begin{aligned} \Delta r &= r_2 - r_1, \\ \Delta z &= z_2 - z_1, \\ \bar{r} &= \frac{1}{2} (r_2 + r_1), \\ \bar{z} &= \frac{1}{2} (z_2 + z_1), \\ \ell &= \sqrt{\Delta r^2 + \Delta z^2}, \\ [H_{qp}^n] &= \frac{1}{\Delta z} \begin{bmatrix} \frac{z_2}{r_1^n} & -\frac{z_1}{r_2^n} \\ -\frac{1}{r_1^n} & \frac{1}{r_2^n} \end{bmatrix} \end{aligned} \right\} \quad (23)$$

The nine by three  $[S_v^t]$  matrix for the CAXIF3 element is calculated with the following equations

$$[S_v] = \begin{bmatrix} [S_c^t] \\ \hline [S_e^t] \end{bmatrix} \quad (24)$$

The three by three  $[S_c^t]$  matrix relates three pressures to the three velocities in the basic coordinate system  $V_r, V_\phi, V_z$ .

$$[S_e^t] = -\frac{1}{\rho} \begin{bmatrix} 0 & 1 & 0 \\ \frac{n}{r_c} & n & \frac{nz_c}{r_c} \\ 0 & 0 & 1 \end{bmatrix} [H_{qp}^n] \quad (25)$$

where  $[H_{qp}^n]$  is a three by three transformation matrix between pressures and generalized coordinates defined in Section 4.87.15.3.

The six by three matrix,  $[S_e^t]$ , which defines the velocities at each edge, tangential and circumferential, is:

# MODULE FUNCTIONAL DESCRIPTIONS

$$[S_e^t] = \frac{1}{\rho} \begin{bmatrix} \frac{1}{l_{12}} & -\frac{1}{l_{12}} & 0 \\ \frac{n}{r_1+r_2} & \frac{n}{r_1+r_2} & 0 \\ 0 & \frac{1}{l_{23}} & -\frac{1}{l_{23}} \\ 0 & \frac{n}{r_2+r_3} & \frac{n}{r_2+r_3} \\ -\frac{1}{l_{13}} & 0 & \frac{1}{l_{13}} \\ \frac{n}{r_1+r_3} & 0 & \frac{n}{r_1+r_3} \end{bmatrix} \quad (26)$$

where  $l_{ij} = \sqrt{(r_j - r_i)^2 + (z_j - z_i)^2}$

The CAXIF4 element is composed of four overlapping triangles. For each triangle I, II, III or IV the connected points 1, 2, 3, 4 are allocated as follows:

| Triangles | Connected points a, b, c |
|-----------|--------------------------|
| I         | 1 2 3                    |
| II        | 1 2 4                    |
| III       | 1 3 4                    |
| IV        | 2 3 4                    |

For each triangle calculate the 3x3  $[S_c^t]$  matrix from Equation 9 and add each column to one of four columns corresponding to the connected point. The results are divided by 4 to provide an average  $[S_c^q]$  matrix for the quadrilateral.

The  $[S_e^q]$  matrix for the quadrilateral is:

# STRUCTURAL ELEMENT DESCRIPTIONS

$$[S_e^q] = -\frac{1}{\rho} \begin{bmatrix} \frac{1}{l_{12}} & -\frac{1}{l_{12}} & & & & \\ \frac{n}{r_1+r_2} & \frac{n}{r_1+r_2} & & & & \\ & \frac{1}{l_{23}} & -\frac{1}{l_{23}} & & & \\ & \frac{n}{r_2+r_3} & \frac{n}{r_2+r_3} & & & \\ & & \frac{1}{l_{34}} & -\frac{1}{l_{34}} & & \\ & & \frac{n}{r_3+r_4} & \frac{n}{r_3+r_4} & & \\ -\frac{1}{l_{41}} & & & & \frac{1}{l_{41}} & \\ \frac{n}{r_4+r_1} & & & & \frac{n}{r_4+r_1} & \end{bmatrix} \quad (27)$$

where  $l_{ij} = \sqrt{(r_j - r_i)^2 + (z_j - z_i)^2}$ .

The resulting  $[S_v]$  matrix for the quadrilateral CAXIF4 element is:

$$[S_v] = \begin{bmatrix} S_c^q \\ \hline S_e^q \end{bmatrix} \quad (28)$$



## MODULE FUNCTIONAL DESCRIPTIONS

### 4.87.15.7 Stress Calculations for the AXIF Elements, Phase 2.

The element identification number, the indices of the connected points, and the  $[S_v]$  matrices are given in the ESTB table. The pressures at the connected points,  $\{P_i\}$ , are given in the UGV matrix data block. Depending on the rigid format, the pressure values are either real or complex numbers and associated with each vector of pressures is a real eigenvalue,  $\lambda$ ; a frequency,  $f$ , or a complex eigenvalue,  $P$ . The equation for velocity is

$$\{V\} = \frac{1}{\omega} [S_v] P_i \quad (29)$$

where  $\{V\}$  is the vector of velocities in the element.

$\omega = \sqrt{|\lambda|}$  (real) in Rigid Format 3 ( $\{P_i\}$  is real)

$\omega = 2\pi f$  (real) in Rigid Formats 8 and 11 ( $\{P_i\}$  is complex)

$\omega = p$  (complex) in Rigid Formats 7 and 10 ( $\{P_i\}$  is complex)

$\omega = 1.0$  in all other Rigid Formats ( $\{P_i\}$  is real)

and  $[S_v]$  is dimensioned 4x2, 9x3, or 11x4 for the CAXIF2, CAXIF3, and CAXIF4 elements respectively.

### 4.87.16 The SLØT3 and SLØT4 Fluid Elements

#### 4.87.16.1 Input Data For the SLØT3 and SLØT4 Elements

1. The ECPT/EST entries for the SLØT3 are:

| <u>Symbol</u>              | <u>Description</u>  |
|----------------------------|---|
| $SIL_1, SIL_2, SIL_3$      | Scalar indices for the connected grid points                            |
| $r_i, z_i, w_i, i = 1,2,3$ | Radius and axis location and slot width of connected grid points, $i$ . |
| $\rho$                     | Density   |
| $B$                        | Bulk Modulus  |
| $M$                        | Number of Slots   |
| $N$                        | Harmonic Number   |

2. ECPT/EST entries for the SLØT4 are the same as for the SLØT3 except four points are used.

## STRUCTURAL ELEMENT DESCRIPTIONS

### 4.87.16.2 General Calculations for the SLØT Elements

1. The overall factor for the number of slots is:

$$F = M, \quad 2N = 0, M, 2M, 3M, \dots \quad (1)$$

$$F = \frac{M}{2}, \quad 2N \neq 0, M, 2M, 3M, \dots \quad (2)$$

2. The SLØT4 element is composed of four overlapping triangles. If the SLØT4 element is used, its data is rearranged to the SLØT3 format and the following operations are carried out for all four subtriangles. A test is made on the direction of the vector normal to the surface of all four triangles if the number of negative normal vectors (NNEG) is one or three, a valid quadrilateral is impossible and a fatal error is set.

### 4.87.16.3 Stiffness Matrix Generation for the SLØT3 Elements

1. For each triangle the following terms are calculated:

$$2 \cdot A = A_2 = [r_1(z_2 - z_3) + r_2(z_3 - z_1) + r_3(z_1 - z_2)] \quad (3)$$

$$C_0 = \frac{F}{6\rho|A_2|} [w_1 + w_2 + w_3] \quad (4)$$

$$\left. \begin{aligned} F_{ir} &= (r_k - r_j) \\ F_{iz} &= (z_j - z_k) \end{aligned} \right\} (i,j,k) = \begin{matrix} (1,2,3) \\ (2,3,1) \\ (3,1,2) \end{matrix} \quad (5)$$

2. The stiffness matrix terms are:

$$K_{ij} = C_0 [F_{ij} F_{jr} + F_{iz} F_{jz}] \quad (6)$$

where

$i$  = the "pivot point"

$j$  = 1, 2, 3

## MODULE FUNCTIONAL DESCRIPTIONS

### 4.87.16.4 Mass Matrix Generation for the SLØT3 Elements

1. The following coefficients are generated:

$$2 \cdot A = A_2 = (r_2 - r_1)(z_3 - z_1) - (r_3 - r_1)(z_2 - z_1) \quad (7)$$

$$\bar{w} = w_1 + w_2 + w_3 + w_i, \quad (8)$$

where i is the "pivot point"

$$C_o = \frac{F|A_2|}{120 B}$$

2. The mass matrix terms are:

$$M_{ij} = C_o(\bar{w} + w_j) \quad j = 1, 2, 3 \neq i$$

$$M_{ij} = 2C_o(\bar{w} + w_j) \quad j = i$$

where i is the "pivot point".

### 4.87.16.5 Stress Matrix Calculations in the SLØT Elements (Phase 1)

The velocities in the SLØT elements are calculated in the same manner as stresses in a structural element. Phase 1 involves calculating pressure field - velocity matrices of the fluid passing through the element.

1. The data placed on the ESTB file are:

$Id_e$  - element identification number

$SIL_1, SIL_2, \dots, SIL_i$  - scalar indices

$[S_v]$  - matrix relation between pressure and velocity.

2. The  $[S_v]$  matrix for the CSLØT3 element is a five by three matrix given as follows:



# STRUCTURAL ELEMENT DESCRIPTIONS

$$[S_V] = -\frac{1}{\rho} \begin{bmatrix} \frac{z_2 - z_3}{A} & \frac{z_3 - z_1}{A} & \frac{z_1 - z_2}{A} \\ \frac{r_3 - r_2}{A} & \frac{r_1 - r_3}{A} & \frac{r_2 - r_1}{A} \\ \hline -\frac{1}{\ell_{12}} & \frac{1}{\ell_{12}} & 0 \\ 0 & -\frac{1}{\ell_{23}} & \frac{1}{\ell_{23}} \\ \frac{1}{\ell_{23}} & 0 & -\frac{1}{\ell_{13}} \end{bmatrix} = \begin{bmatrix} S_V^t \\ \hline S_V^e \end{bmatrix} \quad (9)$$

where

$$\ell_{ij} = \sqrt{(r_j - r_i)^2 + (z_j - z_i)^2} \quad (10)$$

$$A = \frac{1}{2} [r_1(z_2 - z_3) + r_2(z_3 - z_1) + r_3(z_1 - z_2)]$$

The five rows of the matrix correspond to the velocities  $V_{rc}$  and  $V_{zc}$  at the centroid in the  $r$  and  $z$  direction and  $V_1, V_2, V_3$  corresponding to velocities along the three edges.

3. The CSLØT4 element is composed of four overlapping triangles. The velocity at the intersection of the triangles is calculated to be the average of the velocities in each sub-triangle. The subtriangles I, II, III and IV are each given three of the four points 1, 2, 3, 4 as in the following chart:

| Triangle Number | Connected Points |   |   |
|-----------------|------------------|---|---|
|                 | a                | b | c |
| I               | 1                | 2 | 3 |
| II              | 1                | 2 | 4 |
| III             | 1                | 3 | 4 |
| IV              | 2                | 3 | 4 |

The  $[S_V^t]$  matrix for each triangle is calculated and each of the three columns is inserted in one of the four corresponding columns in the  $[S_V^q]$  matrix for the quadrilateral. For instance the first column of  $[S_V^t]$  for triangle IV is inserted in column 2 of the  $[S_V^q]$  matrix. Rows four through seven of the  $[S_V^q]$  matrix are recalculated to correspond to the sides of the

# MODULE FUNCTIONAL DESCRIPTIONS

quadrilateral. The resulting matrix for the CSLØT4 element is:

$$[S_V^q] = \begin{bmatrix} \text{First 2 rows} = \frac{1}{4} \sum [S_V^t] \\ \hline \frac{1}{\rho l_{12}} & -\frac{1}{\rho l_{12}} & & \\ & \frac{1}{\rho l_{23}} & -\frac{1}{\rho l_{23}} & \\ & & \frac{1}{\rho l_{34}} & -\frac{1}{\rho l_{34}} \\ -\frac{1}{\rho l_{41}} & & & \frac{1}{\rho l_{41}} \end{bmatrix} \quad (11)$$

where

$$l_{ij} = \sqrt{(r_j - r_i)^2 + (z_j - z_i)^2}$$

## 4.87.16.6 CSLØTi Element, Phase 2

The data calculated above are extracted from the ESTB data file and the corresponding pressures,  $p_i$ , are extracted from the UGV data block. Associated with each vector is a real or complex number. The general equation for velocity in the element is:

$$\{V\} = \frac{1}{\omega} [S_V] \{p_i\} \quad (12)$$

where  $\{V\}$  is the vector of velocities in the element

$\omega = \sqrt{|\lambda|}$  (real) in Rigid Format 3 ( $\{p_i\}$  is real)

$\omega = 2\pi f$  (real) in Rigid Formats 8 and 11 ( $\{p_i\}$  is complex)

$\omega = p$  (complex) in Rigid Formats 7 and 10 ( $\{p_i\}$  is complex)

$\omega = 1.0$  (real) in all other Rigid Formats ( $\{p_i\}$  is real)

and  $[S_V]$  has dimensions 7x3 or 8x4 for the CSLØT3 and CSLØT4 elements respectively

## STRUCTURAL ELEMENT DESCRIPTIONS

### 4.87.17 Solid Polyhedra Elements, TETRA, WEDGE, HEXA1, HEXA2

These elements define three-dimensional shapes with four points defining a tetrahedron (TETRA), six points defining a wedge (WEDGE), and eight points defining a hexahedron (HEXA1 or HEXA2). Constant strain and stress is assumed in each tetrahedron. The wedge and hexahedron elements are automatically fabricated from tetrahedron elements.

#### 4.87.17.1 Input Data for the Solid Polyhedra Elements

1. The ECPT entries for the solid elements are:

| <u>Symbol</u>  | <u>Description</u>   |
|--|--|
| $I_d$  | Element identification number  |
| M  | Material identification number   |
| $SIL_i, i = 1, N$  | Scalar indices of connected grid points.<br>N = 4, 6, or 8   |
| $CS_i, X_i, Y_i, Z_i \left. \vphantom{\begin{matrix} CS_i, X_i, Y_i, Z_i \\ i = 1, N \end{matrix}} \right\} \\ i = 1, N$ | Coordinate system identification number and<br>location in basic coordinates of connected<br>grid points |
| $\bar{T}$  | Average element temperature  |

2. Coordinate System Data

The numbers  $CS_i, X_i, Y_i,$  and  $Z_i$  are used to calculate  $3 \times 3$  global-to-basic transformation matrices  $[T_i]$  for the connected points. Subroutine TRANSD or TRANSS is used.

3. Material Data

Subroutine MAT is used to generate the following material coefficients:

|          |                               |
|----------|-------------------------------|
| E        | Modulus of elasticity         |
| G        | Shear modulus                 |
| $\nu$    | Poisson's ratio               |
| $\rho$   | Mass density                  |
| $\alpha$ | Thermal expansion coefficient |
| $T_0$    | Reference temperature         |



## MODULE FUNCTIONAL DESCRIPTIONS

### 4.87.17.2 Basic Equations for the TETRA Element

1. The matrix which transforms generalized displacements to grid point displacements is  $[H_{qu}]$  where

$$[H_{uq}] = \begin{bmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ 1 & x_3 & y_3 & z_3 \\ 1 & x_4 & y_4 & z_4 \end{bmatrix} \quad (1)$$

This matrix is inverted to produce the matrix  $[H_{qu}] = [H_{uq}]^{-1}$  and the determinant,  $D$ , of  $[H_{uq}]$ .

The value of the determinant is checked to see if it is consistent with the determinants of the other tetrahedra being used in a single element.

2. The material coefficients  $E$ ,  $G$ , and  $\nu$  are used to generate the  $6 \times 6$  matrix  $[G]$  where the nonzero terms are:

$$G_{11} = G_{22} = G_{33} = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \quad (2)$$

$$G_{12} = G_{21} = G_{13} =$$

$$G_{31} = G_{23} = G_{32} = \frac{E\nu}{(1+\nu)(1-2\nu)} \quad (3)$$

$$G_{44} = G_{55} = G_{66} = G \quad (4)$$

3. The four  $6 \times 3$  matrices  $[C^i]$  which transform displacements at points to strains are generated using elements of the  $H_{qu}$  matrix:  $H_{11}$ ,  $H_{12}$ , etc. The equation is:

# STRUCTURAL ELEMENT DESCRIPTIONS

$$[C^i] = \begin{bmatrix} H_{2i} & 0 & 0 \\ 0 & H_{3i} & 0 \\ 0 & 0 & H_{4i} \\ 0 & H_{4i} & H_{3i} \\ H_{3i} & 0 & H_{1i} \\ H_{2i} & H_{1i} & 0 \end{bmatrix} . \quad (5)$$

## 4.87.17.3 Stiffness Matrix Generations for the TETRA Element (Subroutine KTETRA of Module SMA1)

The 3 x 3 partition of the element stiffness matrix (in global coordinates) connecting points i and j is:

$$[K_{ij}] = [T_i]^T [C^i]^T [G] [C^j] [T_j] ,$$

where  $[T_i]$  and  $[T_j]$  are the 3 x 3 global-to-basic transformation matrices. The matrices are produced for point j corresponding to the pivot point in the matrix assembly process.

## 4.87.17.4 Mass Matrix Generation for the TETRA Element (Subroutine MSØLID of Module SMA2)

The mass matrix for each point of the tetrahedron is formed as a 6 x 6 matrix and inserted on the diagonal of the overall mass matrix. Its equation is:

$$M_{ii} = \begin{bmatrix} m/4 & & & & & \\ & m/4 & & & & \\ & & m/4 & & & \\ & & & m/4 & & \\ & & & & 0 & \\ -0- & & & & & 0 \\ & & & & & & 0 \end{bmatrix} , \quad (6)$$

where  $m = 1/6 \rho |D|$ . (D is the determinant of the  $[H_{uq}]$  matrix.)

## 4.87.17.5 Thermal Load Generation for the TETRA Element (Subroutine TETRA of Module SSG1)

The 3 x 1 thermal load vector  $\{P_i\}$  for point i of the tetrahedron is:

# MODULE FUNCTIONAL DESCRIPTIONS

$$\{P_i\} = [T_i]^T [C_i]^T [G] \{\epsilon_t\} , \quad (7)$$

$$\text{where } \{\epsilon_t\} = \begin{Bmatrix} \alpha \\ \alpha \\ \alpha \\ 0 \\ 0 \\ 0 \end{Bmatrix} (\bar{T} - T_0) , \quad (8)$$

and  $\bar{T} = 1/4 (T_1 + T_2 + T_3 + T_4)$  is the average temperature of the four connected points, given in data block GPTT.

## 4.87.17.6 Stress Calculations for the TETRA Elements (Subroutines SSØLID1 and SSØLID2 of Module SPR2)

The stress is calculated in two phases. Phase I is used to calculate the transformation matrices between displacements and temperatures to stresses. Phase II uses the actual displacements and temperatures to calculate stresses.

1. In Phase I, the following calculations are performed:

$$[S_i] = [G] [C_i] [T_i] \quad i = 1, 2, 3, 4, \quad (9)$$

$$\{S_t\} = [G] \{\alpha\} , \quad (10)$$

$$\text{where } \{\alpha\} = \begin{Bmatrix} \alpha \\ \alpha \\ \alpha \\ 0 \\ 0 \\ 0 \end{Bmatrix} . \quad (11)$$

2. In Phase II, the  $3 \times 1$  displacement vector,  $\{u_i\}$ , for each point,  $i$ , is extracted from the  $\{u_q\}$  displacement vector and the average temperature,  $\bar{T}$ , is extracted from the GPTT data block. The stresses are calculated as follows:



# STRUCTURAL ELEMENT DESCRIPTIONS

$$\begin{pmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{yz} \\ \tau_{xy} \\ \tau_{xy} \end{pmatrix} = \sum_{i=1}^4 [S_i] \{u_i\} - \{S_t\} (\bar{T} - \tau_0). \quad (12)$$

The hydrostatic pressure,  $P$ , and the octahedral shear stress,  $\tau_0$ , are calculated by the equations:

$$P = -1/3 (\sigma_x + \sigma_y + \sigma_z), \quad (13)$$

$$\tau_0 = 1/3 [2\sigma_x (\sigma_x - \sigma_y - \sigma_z) + 2\sigma_y (\sigma_y - \sigma_z) + 2\sigma_z^2 + 6 (\tau_{yz}^2 + \tau_{xz}^2 + \tau_{xy}^2)]^{1/2} \quad (14)$$

## 4.87.17.7 Basic Equations for the WEDGE, HEXA1, and HEXA2 Elements

The wedge element is connected to six grid points and is divided into four tetrahedron sub-elements. The connected points assigned to each tetrahedron are:

| <u>TETRA Number</u> | <u>Connected Points</u> |   |   |   |
|---------------------|-------------------------|---|---|---|
| I                   | 1                       | 2 | 3 | 6 |
| II                  | 1                       | 2 | 6 | 5 |
| III                 | 1                       | 4 | 5 | 6 |

The HEXA1 and HEXA2 elements are connected to eight grid points and are subdivided into five tetrahedra for the HEXA1 element and ten overlapping tetrahedra for the HEXA2 element. The connected points original to each tetrahedron are:

# MODULE FUNCTIONAL DESCRIPTIONS

| <u>Subelement Number</u> |       | <u>Connected Points</u> |   |   |   |
|--------------------------|-------|-------------------------|---|---|---|
| HEXA1                    | HEXA2 |                         |   |   |   |
| I                        | I     | 1                       | 2 | 3 | 6 |
| II                       | II    | 1                       | 3 | 4 | 8 |
| III                      | III   | 1                       | 3 | 8 | 6 |
| IV                       | IV    | 1                       | 5 | 6 | 8 |
| V                        | V     | 3                       | 6 | 7 | 8 |
|                          | VI    | 2                       | 3 | 4 | 7 |
|                          | VII   | 1                       | 2 | 4 | 5 |
|                          | VIII  | 2                       | 4 | 5 | 7 |
|                          | IX    | 2                       | 5 | 6 | 7 |
|                          | X     | 4                       | 5 | 7 | 8 |

The basic procedure used with these elements is to extract the data associated with each tetrahedron subelement and go to the tetrahedron calculations. In subroutine KSØLID of Module SMA1, the tetrahedron calculations and matrix insertion is done by calling subroutine KTETRA. In subroutine MSØLID of Module SMA2, the tetrahedron calculations and insertion are done in an internal subroutine. In subroutine SØLID of Module SSG1, the tetrahedron subroutine STETRA is used to calculate and invert the thermal loads. In subroutines SSØLID1 and SSØLID2 of Module SDR2, the tetrahedron calculations are done with an internal subroutine and the results are summed together to produce average stresses.

## 4.87.17.8 Stiffness Matrix Calculations and Geometry Checks for the WEDGE, HEXA1, and HEXA2 Elements (Subroutine KSØLID of Module SMA1)

With these elements, the order of the connections and the resulting geometry is critical for reasonable results. Three basic criteria must be met:

1. If the connections are correct, the calculated volumes for all tetrahedron subelements will be consistent. When the subroutine is called for the first time for each element, all of the tetrahedra are processed to produce the signs of the determinants of the  $[H_{uq}]$  matrices. The signs must be either all positive or all negative, or an error is indicated.

## STRUCTURAL ELEMENT DESCRIPTIONS

2. The order of the connected points is checked by calculating the normal vectors to the top and bottom faces assuming a right-hand rule. The normal vectors must not have a negative scalar product.
3. The wedge has three quadrilateral faces and the hexagonal elements have six quadrilateral faces. Subroutine KPLTST is used to check these faces. The points must not deviate from being a plane by more than 10 percent.

### Wedge

| Face Number | Points on Face |
|-------------|----------------|
| 1           | 1, 2, 5, 4     |
| 2           | 1, 4, 6, 3     |
| 3           | 2, 3, 6, 5     |

### Hexahedron

| Face Number | Points on Face |
|-------------|----------------|
| 1           | 1, 2, 3, 4     |
| 2           | 1, 2, 6, 5     |
| 3           | 2, 3, 7, 6     |
| 4           | 3, 4, 8, 7     |
| 5           | 4, 1, 5, 8     |
| 6           | 5, 6, 7, 8     |

In the KSØLID subroutine, each element is tested for geometric consistency when the pivot point equals the first connected point. In any event, the ECPT data is converted to the TETRA format for as many times as necessary, and subroutine KTETRA is called each time. If a HEXA2 element is being processed, a flag is set, so the KTETRA subroutine will divide the stiffness of each tetrahedron by two.

#### 4.87.17.9 Mass Matrix Generation for the WEDGE, HEXA1 and HEXA2 Elements (Subroutine MSØLID of Module SMA2)

The mass calculations involve the calculation of the total mass of each tetrahedron in the element and assigning one-fourth of the mass to each of the four points. If a HEXA2 element is used, the mass of each tetrahedron is divided by two.



## MODULE FUNCTIONAL DESCRIPTIONS

### 4.87.17.10 Thermal Load Generation for the WEDGE, HEXA1 and HEXA2 Elements (Subroutine SØLID of Module SSG2)

This subroutine arranges the ECPT data into the TETRA format for each tetrahedron in the element. Subroutine TETRA is called each time to calculate the thermal loads and insert them in the load vector. If a HEXA2 element is used, a flag is set, so that the TETRA routine will divide the results by two.

### 4.87.17.11 Stress Data Recovery for the WEDGE, HEXA1 and HEXA2 Elements (Subroutines SSØLID1 and SSØLID2 of Module SDR2)

The first phase of stress recovery involves the calculation of displacement-stress matrices  $[S_{ie}]$  and the temperature stress vector  $\{S_{te}\}$ . A  $6 \times 3$   $[S_{ie}]$  matrix is generated for each connected point. Its equation is:

$$[S_{ie}] = \frac{1}{N} \sum_{\alpha=1}^N [S_i]_{\alpha}, \quad (15)$$

where  $[S_i]_{\alpha}$  is the matrix corresponding to tetrahedron number  $\alpha$  associated with point  $i$ , and  $N$  is the total number of tetrahedra in the element. The  $[S_i]_{\alpha}$  matrices are described in Section 4.87.17.6. As each tetrahedron is processed, the four  $[S_i]$  matrices and the  $\{S_t\}$  vector are added to the appropriate  $[S_{ie}]$  matrices for the whole element. The TETRA element is also processed by this code with  $N = 1$ . The thermal stress vectors are added by the equation:

$$\{S_{te}\} = \frac{1}{N} \sum_{\alpha=1}^N \{S_t\}_{\alpha} \quad (16)$$

In Phase II of stress recovery, the logic is given in Section 4.87.17.6. The code is identical for all elements, with the only difference being the number of connected grid points.

## STRUCTURAL ELEMENT DESCRIPTIONS

### 4.87.17.12 Thermal Analysis Calculations for the Solid Elements

The "stiffness" matrix terms for the solid elements are generated by subroutines KSØLID and KTETRA of module SMA1. All of the solid elements (TETRA, WEDGE, HEXA1, and HEXA2) use the KTETRA subroutine to calculate and insert the final matrix terms. For thermal analysis, the following operations are performed:

1. The geometry is processed and the matrix  $[H_{uq}]$  and the determinant,  $D$ , are produced for either structure or thermal analysis. See Section 4.87.17.3.
2. For thermal analysis, subroutine HMAT is used with INFLAG = 3 to produce the following data:

$$K_{xx} \ K_{xy} \ K_{xz} \ K_{yy} \ K_{yz} \ K_{zz}$$

3. The material matrix  $[G_e]$  is calculated where:

$$[G_e] = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & K_{xx} & K_{xy} & K_{xz} \\ 0 & K_{xy} & K_{yy} & K_{yz} \\ 0 & K_{xz} & K_{yz} & K_{zz} \end{bmatrix}$$

4. The matrix terms associated with the pivot point (j) are:

$$\begin{aligned} & \{ \bar{K}_{j1} \ \bar{K}_{j2} \ \bar{K}_{j3} \ \bar{K}_{j4} \}^T \\ & = \frac{D}{6} \{ H_{uq}^j \}^T [G_e] [H_{uq}]. \end{aligned}$$

The vector  $\{ H_{uq}^j \}$  is the column of the  $[H_{uq}]$  matrix associated with point j. The values  $K_{ji}$  are inserted in the KGG matrix in column number =  $SIL_j$ , and row number =  $SIL_i$ .

The "mass" matrix for the heat transfer analysis is generated by subroutine MSØLID of module SMA2. The thermal capacity coefficient,  $C_p$ , is returned from subroutine HMAT. Each subelement tetrahedron is used to calculate terms, which are placed in the BGG matrix. For each point, i, on each tetrahedron, the value is:

$$B_{ii} = \frac{C_p \cdot |D|}{6},$$

where  $D$  is the determinant of the  $[H_{uq}]$  matrix. Results for the HEXA2 element are divided by 2.



## MODULE FUNCTIONAL DESCRIPTIONS

The "stress" data recovery for heat transfer analysis is performed by subroutines SDHTF1, SDHTFF, and SDHTF2 of module SDR2. In Phase 1, the matrix K and the matrix C are calculated where K is the 3 x 3 material matrix and  $C_e$  is a 3 by number of points matrix. For the tetrahedron:

$$[C_e] = \begin{bmatrix} H_{21} & H_{22} & H_{23} & H_{24} \\ H_{31} & H_{32} & H_{33} & H_{34} \\ H_{41} & H_{42} & H_{43} & H_{44} \end{bmatrix}$$

For the WEDGE, HEXA1, and HEXA2 elements, the [C] matrix is calculated for each subelement. The column corresponding to each point of the tetrahedron is added to the column of the  $C_e$  matrix corresponding to that point in the whole element. The results are divided by the number of subelements.

In Phase 2, the temperature gradient vector and flux vector are calculated with the equations:

$$\{\Delta T\} = [C] \{u\}$$

$$\{q\} = -[K]\{\Delta T\},$$

where  $\{u\}$  is the vector of temperatures of the connected points.



## STRUCTURAL ELEMENT DESCRIPTIONS

### 4.87.18 The HBDY Elements

#### 4.87.18.1 Input Data

The boundary condition element HBDY produces matrix terms only in a heat transfer analysis. The following data will be needed from the ECPT table.

| <u>Symbol</u>                                 | <u>Description</u>  |
|---|---|
| ID  | Element ID  |
| IFLAG   | Element Type Flag   |
| $SIL_i, i = 1, 2 \dots 8$                     | Scalar indices  |
| $V_1, V_2, V_3$                               | Orientation vector  |
| Mat Id  | Material identification number for MAT4 or MAT5 data  |
| AF  | Area factor   |
| $\epsilon$                                    | Emissivity coefficient  |
| $\alpha$                                      | Absorbtivity coefficient  |
| $R_1, R_2$                                    | Radii of elliptic cylinder  |
| $C_{si}, X_i, Y_i, Z_i$<br>$i = 1, 2 \dots 8$ | Coordinate systems and location coordinates of connected grid points; 1 - 4 are element points, 5 - 8 are points in fluid |
| To  | Average element temperature for initial estimate  |

The meaning of the data for various values of IFLAG are:

| IFLAG | Element Type      | N= Number of Gridpoints | AF            |
|-------|-------------------|-------------------------|---------------|
| 1     | point             | 1                       | area          |
| 2     | line              | 2                       | width         |
| 3     | revolution        | 2                       | ---           |
| 4     | triangle          | 3                       | ---           |
| 5     | quadrilateral     | 4                       | ---           |
| 6     | elliptic cylinder | 2                       | circumference |

# MODULE FUNCTIONAL DESCRIPTIONS

## 4.87.18.2 Stiffness Matrix Calculations (Subroutine KHBDY of Module SMA1)

For the revolution elements  $x_i > 0$  and  $y_i = 0$ , otherwise there is illegal geometry. The matrix produced will be  $N \times N$ . The material coefficient,  $H$ , is extracted from the material data block with INFLAG = 1, and  $T_0$  as the average temperature. The  $[C]$  matrix for each element type is given in the following table:

| IFLAG     | C  |
|-----------|--|
| 1         | $H(AF)$  |
| 2<br>or 6 | $\frac{H(AF)\ell}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$   |
| 3         | $\frac{H(2\pi)\ell}{12} \begin{bmatrix} (3x_1 + x_2) & (x_1 + x_2) \\ (x_1 + x_2) & (x_1 + 3x_2) \end{bmatrix}$  |
| 4         | $\frac{Ha}{24} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}$  |
| 5         | <p>or</p> $\frac{H}{48} \begin{bmatrix} 2(a_2 + a_3 + a_4) & (a_3 + a_4) & (a_2 + a_4) & (a_2 + a_3) \\ & 2(a_1 + a_3 + a_4) & (a_1 + a_4) & (a_1 + a_3) \\ & & 2(a_1 + a_2 + a_4) & (a_1 + a_2) \\ & & & 2(a_1 + a_2 + a_3) \end{bmatrix}$ <p>where</p> $C_{ij} = \frac{H}{48} [(1 + \delta_{ij})(a_1 + a_2 + a_3 + a_4) - (a_i + a_j)],$ $\delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$ |

The length  $\ell$  appears only when  $N = 2$  or  $6$ , and:

$$\ell = [(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2]^{1/2}.$$

The factor  $a$  is twice the area of a triangle ( $N = 3$ ).



## STRUCTURAL ELEMENT DESCRIPTIONS

Let

$$\vec{r}_i = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}$$

$$a = |(\vec{r}_2 - \vec{r}_1) \times (\vec{r}_3 - \vec{r}_1)|$$

The factors  $a_i$  are two times the area of the triangle which does not touch vertex  $i$  of a quadrilateral.

$$a_1 = |(\vec{r}_3 - \vec{r}_2) \times (\vec{r}_4 - \vec{r}_2)|$$

$$a_2 = |(\vec{r}_4 - \vec{r}_3) \times (\vec{r}_1 - \vec{r}_3)|$$

$$a_3 = |(\vec{r}_1 - \vec{r}_4) \times (\vec{r}_2 - \vec{r}_1)|$$

$$a_4 = |(\vec{r}_2 - \vec{r}_1) \times (\vec{r}_3 - \vec{r}_2)|$$

The C matrices are placed in the "stiffness" matrix using the following matrix equation:

$$\{P\} = [K] \{u\},$$

or:

$$\begin{pmatrix} P_1 \\ \vdots \\ P_N \\ \hline P_5 \\ \vdots \\ P_{4+N} \end{pmatrix} = \begin{bmatrix} C & | & -C \\ \hline -C & | & C \end{bmatrix} \begin{pmatrix} u_1 \\ \vdots \\ u_N \\ \hline u_5 \\ \vdots \\ u_{4+N} \end{pmatrix}$$

In the octimal code, only the columns corresponding to a "pivot point"  $u_i$  are generated.

The terms  $K_{ji}$  are inserted for every point  $u_i$  in the element SIL list. The terms are placed in the  $SIL_i$  column of the matrix.

### 4.87.18.3 Capacity Matrix Calculations for the HBDY Element (Subroutine MHBDY of Module SMA2)

The heat capacity matrix is calculated much like a mass matrix and is placed in matrix BGG in the SMA2 module. The material coefficient  $C_p$  is extracted from the material tables using subroutine HMAT with INFLAG = 5. The scalar terms for each point are:



# MODULE FUNCTIONAL DESCRIPTIONS

| IFLAG  | B  |
|--------|--|
| 1      | $B_{11} = AF \cdot Cp$   |
| 2 or 6 | $B_{11} = B_{22} = \frac{AF \cdot Cp \cdot \ell}{2}$   |
| 3      | $B_{11} = \frac{\pi Cp \ell}{3} (2x_1 + x_2)$<br>$B_{22} = \frac{\pi Cp \ell}{3} (2x_2 + x_1)$ |
| 4      | $B_{11} = B_{22} = B_{33} = \frac{a_1 \cdot Cp}{6}$  |
| 5      | $B_{ii} = \frac{(a_T - a_i) \cdot Cp}{12} \quad i = 1, 4$                                      |

where  $a_T = a_1 + a_2 + a_3 + a_4$  and the  $a$  values are defined in the previous section.

One-half the value of the terms are placed in the BGG matrix in positions row = column =  $SIL_i$ . If the corresponding boundary layer point  $SIL_{i+4}$  is nonzero, the value,  $\frac{1}{2}B_{ii}$ , is also placed in the row and column corresponding to  $SIL_{i+4}$ .

4.87.18.4 Convective Heat Flux Recovery for the HBDY Element (Subroutines SDHTF1, SDHTFF, and SDHTF2 in Module SDR2)

## Phase 1

Subroutine SDHTF1 rearranges the EST data and calls the HMAT routine for material property, H. The output of Phase 1 consists of the following data:

| Word  | Description                            |
|-------|--|
| 1     | Element Id                             |
| 2-9   | Scalar indices of connected points     |
| 10,11 | BCD words for "HBDY"                   |
| 12    | Number of generalized coordinates (=1) |
| 13    | NP - Number of connected points (=8)   |
| 14    | The value of H                         |
| 22-29 | The eight C values                     |

## STRUCTURAL ELEMENT DESCRIPTIONS

Subroutine SDHTFF calculates the nonzero values of C and the value K with the following equations

$$\text{IFLAG} = 1$$

$$C_1 = -C_5 = 1.0$$

$$K = A_f H$$

$$\text{IFLAG} = 2 \text{ or } 6$$

$$C_1 = C_2 = -C_5 = -C_6 = \frac{1}{2}$$

$$K = \frac{A_f \cdot L \cdot H}{2}$$

$$\text{IFLAG} = 3$$

$$C_1 = \frac{1}{3\bar{r}} (2x_1 + x_2) = -C_5$$

$$C_2 = \frac{1}{3\bar{r}} (2x_2 + x_1) = -C_6$$

$$K = \pi \bar{r} L H,$$

$$\text{where } \bar{r} = \frac{1}{2} (x_1 + x_2)$$

$$\text{IFLAG} = 4$$

$$C_1 = C_2 = C_3 = -C_5 = -C_6 = -C_7 = \frac{1}{3}$$

$$K = AH,$$

$$\text{where } A = \frac{1}{2} |(\bar{r}_2 - \bar{r}_1) \times (\bar{r}_3 - \bar{r}_1)|$$

$$\text{IFLAG} = 5$$

$$C_1 = C_2 = C_3 = C_4 = -C_5 = -C_6 = C_7 = -C_8 = \frac{1}{4}$$

$$K = A_q H, \text{ where } A_q = \frac{1}{2} \sum (\text{area of each triangle})$$

### Phase 2

Subroutine SDHTF2 calculates the temperature difference and total heat flow across the boundary layer with the equation

$$\Delta T = \{C\}^T \{u\}$$

$$q = -K \Delta T,$$



## MODULE FUNCTIONAL DESCRIPTIONS

where  $K$  and  $\{C\}$  are the Phase 1 outputs and  $\{u\}$  is the vector of temperatures given in the UGV vector in the  $SIL^{th}$  positions.

### 4.87.18.5 Area Factor Calculations for the HBDY Element (Utility Subroutine HBDY)

For purposes of radiation calculations, thermal load generation, and data recovery, the HBDY elements are processed by subroutine HBDY. The input data consists of the EST entry for each element. The output is dependent on the option as follows:

a) Option = 1

| <u>Word</u> | <u>Symbol</u> | <u>Description</u>                                |
|-------------|---------------|---|
| 1           | ID            | Element ID  |
| 2           | $A_T$         | Total area  |
| 3           | $\epsilon$    | Emissivity  |
| 4           | Not used      |   |
| 5-8         | $SIL_i$       | Scalar indices                                    |
| 9-12        | $G_i$         | Fraction of total area associated with each point |

b) Option = 2

| <u>Word</u> | <u>Symbol</u> | <u>Description</u>                      |
|-------------|---------------|---|
| 1           | ID            | Element ID                              |
| 2           | $A_T$         | Total area                              |
| 3-6         | $SIL_i$       | Scalar indices                          |
| 7-10        | $A_i$         | Area factors                            |
| 11-13       | $\bar{n}_1$   | Normal unit vector                      |
| 14-16       | $\bar{n}_2$   | Second normal vector for IFLAG = 6 only |

The output quantities are calculated as follows:

$$IFLAG = 1$$

$$\bar{n}_1 = \frac{\bar{v}}{|\bar{v}|}$$

$$A_T = AF$$

$$A_i = AF$$



# STRUCTURAL ELEMENT DESCRIPTIONS

$$\text{IFLAG} = 2 \quad \bar{\mathbf{R}} = \begin{pmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{pmatrix}$$

$$\ell = |\mathbf{R}|$$

$$A_T = A_F \cdot \ell$$

$$A_1 = A_2 = \frac{1}{2} A_T$$

$$\bar{\mathbf{Z}} = \bar{\mathbf{V}} - \frac{(\bar{\mathbf{R}} \cdot \bar{\mathbf{V}})}{\ell^2} \bar{\mathbf{R}}$$

$$\bar{\mathbf{n}}_1 = \frac{\bar{\mathbf{Z}}}{|\bar{\mathbf{Z}}|}$$

$$\text{IFLAG} = 3$$

$$\Delta r = r_2 - r_1$$

$$\Delta z = z_2 - z_1$$

$$\ell = \sqrt{\Delta r^2 + \Delta z^2}$$

$$A_1 = \frac{\pi \ell}{3} (2r_1 + r_2)$$

$$A_2 = \frac{\pi \ell}{3} (2r_2 + r_1)$$

$$A_T = A_1 + A_2$$

$$\bar{\mathbf{n}}_1 = \frac{1}{\ell} \begin{pmatrix} \Delta z \\ 0 \\ -\Delta r \end{pmatrix}$$

$$\text{IFLAG} = 4$$

$$\bar{\mathbf{D}}_1 = \bar{\mathbf{x}}_2 - \bar{\mathbf{x}}_1$$

$$\bar{\mathbf{D}}_2 = \bar{\mathbf{x}}_3 - \bar{\mathbf{x}}_1$$

$$\bar{\mathbf{Z}} = \bar{\mathbf{D}}_1 \times \bar{\mathbf{D}}_2$$

$$A_T = \frac{1}{2} |\bar{\mathbf{Z}}|$$

$$\bar{\mathbf{n}} = \frac{\bar{\mathbf{Z}}}{|\bar{\mathbf{Z}}|}$$

# MODULE FUNCTIONAL DESCRIPTIONS

$$A_1 = A_2 = A_3 = \frac{1}{3} A_T$$

IFLAG = 5

$$\bar{D}_{12} = \bar{r}_2 - \bar{r}_1$$

$$\bar{D}_{13} = \bar{r}_3 - \bar{r}_1$$

$$\bar{D}_{24} = \bar{r}_4 - \bar{r}_2$$

$$\bar{Z} = \bar{D}_{13} \times \bar{D}_{24}$$

$$A_T = \frac{1}{2} |\bar{Z}|$$

$$\bar{n}_1 = \frac{\bar{Z}}{|\bar{Z}|}$$

$$A_{123} = \frac{1}{2} |\bar{D}_{12} \times \bar{D}_{13}|$$

$$A_{412} = \frac{1}{2} |\bar{D}_{24} \times \bar{D}_{12}|$$

$$A_1 = \frac{1}{6} (A_T + A_{412})$$

$$A_2 = \frac{1}{6} (A_T + A_{213})$$

$$A_3 = \frac{1}{6} (2A_T - A_{412})$$

$$A_4 = \frac{1}{6} (2A_T - A_{213})$$

IFLAG = 6

$$\bar{x} = \bar{r}_2 - \bar{r}_1$$

$$\ell = |\bar{x}|$$

$$A_T = A_f \cdot \ell - \text{Option 1}$$

$$\text{or } A_T = \ell - \text{Option 2}$$

$$A_1 = A_2 = \frac{1}{2} A_T$$

$$\bar{y} = \bar{v} \times \bar{x}$$

# STRUCTURAL ELEMENT DESCRIPTIONS

$$\bar{z} = \bar{x} \times \bar{y}$$

$$n_1 = \frac{\bar{z}}{|\bar{z}|} \cdot R_2$$

$$n_2 = \frac{\bar{y}}{|\bar{y}|} \cdot R_1$$

Note for Option 1, the factors  $G_i$  are

$$G_i = \frac{A_i}{A_T}$$



## STRUCTURAL ELEMENT DESCRIPTIONS

### 4.87.19 QDMEM1 Isoparametric Quadrilateral Element

#### 4.87.19.1 Input Data for QDMEM1 Element

1. ECPT entries for QDMEM1 are:

| <u>Symbol</u>   | <u>Description</u>   |
|---|--|
| EID   | Element identification number  |
| $\left. \begin{array}{l} \text{SIL}_1 \\ \text{SIL}_2 \\ \text{SIL}_3 \\ \text{SIL}_4 \end{array} \right\}$ | Scalar indices of the connected grid points  |
| $\theta$  |  |
| Mat ID  |  |
| $h$   |  |
| $\mu$   | Nonstructural mass per unit area   |
| $\left. \begin{array}{l} N_i \\ X_i \\ Y_i \\ Z_i \end{array} \right\} \quad i = 1, 4$                      | Local coordinate system numbers and location coordinates in the basic system for the connected grid points |
| $\bar{t}$   |  |
|   |  |
|   |  |
|   | Temperature of element   |

2. Coordinate system data

The numbers  $N_i$ ,  $X_i$ ,  $Y_i$ , and  $Z_i$  are used to calculate 3 by 3 basic-to-global coordinate transformation matrices  $[T_i]$  for points  $i = 1, 2, 3$ , and 4.

3. Material Data

| <u>Symbol</u>                     | <u>Description</u>                                |
|-----------------------------------|---|
| $[G]$                             | 3x3 Stress-strain matrix                          |
| $\rho$                            | Mass density                                      |
| $\alpha_x, \alpha_y, \alpha_{xy}$ | Thermal expansion coefficients                    |
| $t_0$                             | Reference temperature                             |
| $g_e$                             | Structural damping coefficient                    |
| $\sigma_t, \sigma_c, \sigma_s$    | Stress limits for tension, compression, and shear |

# MODULE FUNCTIONAL DESCRIPTIONS

## 4.87.19.2 Basic Equations for QDMEM1

### 1. Calculation of unit vectors in mean plane.

The following equations are used to calculate the three unit vectors in the mean plane,  $\{i\}$ ,  $\{j\}$ , and  $\{k\}$  (see figure 1), which define the element coordinate system.

$$\{V_i\} = \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix}, \quad i = 1, 2, 3, 4 \quad (1)$$

The diagonals are:

$$\{d_1\} = \{V_3\} - \{V_1\} \quad (2)$$

$$\{d_2\} = \{V_4\} - \{V_2\} \quad (3)$$

The normal to the plane is calculated from:

$$\{k\} = \frac{\{d_1\} \times \{d_2\}}{|\{d_1\} \times \{d_2\}|} \quad (4)$$

$$\{a_1\} = \{V_2\} - \{V_1\} \quad (5)$$

$$\hat{h} = \frac{1}{2} \{a_1\}^T \{k\} \quad (6)$$

The vectors lying in the mean plane are computed from:

$$\{i\} = \frac{\{a_1\} - 2\hat{h}\{k\}}{|\{a_1\} - 2\hat{h}\{k\}|} \quad (7)$$

$$\{j\} = \{k\} \times \{i\} \quad (8)$$

### 2. The displacement transformation matrix from basic coordinates to in-plane coordinates is:



# STRUCTURAL ELEMENT DESCRIPTIONS

$$[E]^T = \begin{bmatrix} \bar{E}^T & 0 & 0 & 0 \\ 0 & \bar{E}^T & 0 & 0 \\ 0 & 0 & \bar{E}^T & 0 \\ 0 & 0 & 0 & \bar{E}^T \end{bmatrix} \quad (9)$$

where

$$[\bar{E}]^T = \begin{bmatrix} i_1 & i_2 & i_3 \\ j_1 & j_2 & j_3 \\ k_1 & k_2 & k_3 \end{bmatrix} \quad (10)$$

3. The coordinates or difference of coordinates of the points in the element coordinate system are:

$$x_{12} = x_1 - x_2 = -\{a_1\}^T\{i\} \quad (11)$$

$$x_{13} = x_1 - x_3 = -\{d_1\}^T\{i\} \quad (12)$$

$$x_{24} = x_2 - x_4 = -\{d_2\}^T\{i\} \quad (13)$$

$$x_{14} = x_{12} + x_{24} \quad (14)$$

$$x_{23} = x_{13} - x_{12} \quad (15)$$

$$x_{34} = x_{14} - x_{13} \quad (16)$$

$$y_3 = \{d_1\}^T\{j\} \quad (17)$$

$$y_4 = \{d_2\}^T\{j\} \quad (18)$$

$$y_{34} = y_3 - y_4 \quad (19)$$



# MODULE FUNCTIONAL DESCRIPTIONS

4. The transformation of displacements at the user-specified grid points from the displacements at the projected points in the mean plane is given by the  $12 \times 8$  matrix [B]. The nonzero elements of this matrix are given by:

$$B(1,1) = B(2,2) = B(4,3) = B(5,4) = B(7,5) = B(8,6) = B(10,7) = B(11,8) = 1 \quad (20)$$

$$B(3,1) = \frac{-\bar{h}}{\ell_a} \quad (21)$$

$$B(3,2) = \bar{h} \left( \frac{-1}{\ell_d \sin \theta_1} + \frac{\cot \theta_1}{\ell_a} \right) \quad (22)$$

$$B(3,3) = -B(3,1) \quad (23)$$

$$B(3,4) = \bar{h} \left( \frac{\cot \theta_2}{\ell_a} \right) \quad (24)$$

$$B(3,7) = \bar{h} \left( \frac{\cos \theta_{42}}{\ell_d \Delta_2} \right) \quad (25)$$

$$B(3,8) = \bar{h} \left( \frac{\sin \theta_{42}}{\ell_d \Delta_2} \right) \quad (26)$$

$$B(6,1) = -B(3,1) \quad (27)$$

$$B(6,2) = -\bar{h} \left( \frac{\cot \theta_1}{\ell_a} \right) \quad (28)$$

$$B(6,3) = B(3,1) \quad (29)$$

$$B(6,4) = \bar{h} \left( \frac{-\cot \theta_2}{\ell_a} + \frac{1}{\ell_b \sin \theta_2} \right) \quad (30)$$

$$B(6,5) = \bar{h} \left( \frac{-\sin \theta_{31}}{\ell_b \Delta_1} \right) \quad (31)$$

$$B(6,6) = \bar{h} \left( \frac{-\cos \theta_{31}}{\ell_b \Delta_1} \right) \quad (32)$$

# STRUCTURAL ELEMENT DESCRIPTIONS

$$B(9,4) = \bar{h} \left( \frac{-1}{\ell_b \sin \theta_2} \right) \quad (33)$$

$$B(9,5) = \bar{h} \left( \frac{\sin \theta_{31}}{\ell_b \Delta_1} + \frac{\cos \theta_{32}}{\ell_c \Delta_1} \right) \quad (34)$$

$$B(9,6) = \bar{h} \left( \frac{\cos \theta_{31}}{\ell_b \Delta_1} + \frac{\sin \theta_{32}}{\ell_c \Delta_1} \right) \quad (35)$$

$$B(9,7) = \bar{h} \left( \frac{-\sin \theta_{41}}{\ell_c \Delta_2} \right) \quad (36)$$

$$B(9,8) = \bar{h} \left( \frac{\cos \theta_{41}}{\ell_c \Delta_2} \right) \quad (37)$$

$$B(12,2) = \bar{h} \left( \frac{1}{\ell_d \sin \theta_1} \right) \quad (38)$$

$$B(12,5) = \bar{h} \left( \frac{-\cos \theta_{32}}{\ell_c \Delta_1} \right) \quad (39)$$

$$B(12,6) = \bar{h} \left( \frac{-\sin \theta_{32}}{\ell_c \Delta_1} \right) \quad (40)$$

$$B(12,7) = \bar{h} \left( \frac{-\cos \theta_{42}}{\ell_d \Delta_2} + \frac{\sin \theta_{41}}{\ell_c \Delta_2} \right) \quad (41)$$

$$B(12,8) = \bar{h} \left( \frac{-\sin \theta_{42}}{\ell_d \Delta_2} - \frac{\cos \theta_{41}}{\ell_c \Delta_2} \right) \quad (42)$$

where

$$\bar{h} = -\hat{h} \quad (43)$$

$$\ell_a = -x_{12} \quad (44)$$

# MODULE FUNCTIONAL DESCRIPTIONS

$$l_b = \sqrt{x_{23}^2 + y_3^2} \quad (45)$$

$$l_c = \sqrt{x_{34}^2 + y_{34}^2} \quad (46)$$

$$l_d = \sqrt{x_{14}^2 + y_4^2} \quad (47)$$

$$\sin \theta_1 = \sin \theta_{41} = \frac{y_4}{l_d} \quad (48)$$

$$\cos \theta_1 = \cos \theta_{41} = \frac{-x_{14}}{l_d} \quad (49)$$

$$\sin \theta_2 = \cos \theta_{32} = \frac{y_3}{l_b} \quad (50)$$

$$\cos \theta_2 = \sin \theta_{32} = \frac{x_{23}}{l_b} \quad (51)$$

$$\sin \theta_{31} = \cos \theta_{42} = -\frac{y_{34}}{l_c} \quad (52)$$

$$\cos \theta_{31} = \sin \theta_{42} = \frac{x_{34}}{l_c} \quad (53)$$

$$\Delta_1 = \cos \theta_{31} \cos \theta_{32} - \sin \theta_{31} \sin \theta_{32} \quad (54)$$

$$\Delta_2 = \cos \theta_{41} \cos \theta_{42} + \sin \theta_{41} \sin \theta_{42} \quad (55)$$

See Figure 2 for characteristic lengths and angles of quadrilateral membrane in mean plane.



## 4.87.19.3 Stiffness Matrix Calculation for QDMEM1 (Subroutine KQDMM1 of Module SMA1)

## 1. Element interior angle tests.

The interior angles of the quadrilateral in the mean plane are checked to verify that they are less than  $180^\circ$ . The following tests are used:

| Test  | Point with angle greater than $180^\circ$ |
|---|---|
| If $y_4 < 0$                                  | 1   |
| If $y_3 < 0$                                  | 2   |
| If $x_{14} < x_{12} + x_{23} \frac{y_4}{y_3}$ | 3   |
| If $x_{13} > \frac{y_3}{y_4} x_{14}$          | 4   |

## 2. The stiffness matrix in the mean plane in terms of the displacements in the mean plane is calculated first as

$$[U]_{8 \times 8} = \int_0^1 \int_0^1 [A]^T [G] [A] J d\xi d\eta \quad (56)$$

where  $[G]$  is the  $3 \times 3$  stress-strain matrix,

$$J = -x_{12}y_4 - x_{12}y_{34} \xi + (y_3x_{14} - y_4x_{23})\eta \quad (57)$$

and  $[A]$ , a  $3 \times 8$  strain-displacement matrix is represented in the following manner.

$$[A]_{3 \times 8} = \frac{1}{J} \begin{bmatrix} (a_1 + b_1\eta + c_1\xi) & (a_4 + b_4\eta + c_4\xi) & . & . & . & (a_{22} + b_{22}\eta + c_{22}\xi) \\ (a_2 + b_2\eta + c_2\xi) & . & . & . & . & (a_{23} + b_{23}\eta + c_{23}\xi) \\ (a_3 + b_3\eta + c_3\xi) & . & . & . & . & (a_{24} + b_{24}\eta + c_{24}\xi) \end{bmatrix} \quad (58)$$

In the expression for the matrix  $[A]$ ,

$$a_1 = -y_4 \quad b_1 = y_3 \quad c_1 = -y_{34} \quad (59)$$

$$a_2 = 0 \quad b_2 = 0 \quad c_2 = 0 \quad (60)$$

# MODULE FUNCTIONAL DESCRIPTIONS

$$a_3 = -x_{24} \quad b_3 = x_{23} \quad c_3 = x_{34} \quad (61)$$

$$a_4 = 0 \quad b_4 = 0 \quad c_4 = 0 \quad (62)$$

$$a_5 = -x_{24} \quad b_5 = x_{23} \quad c_5 = x_{34} \quad (63)$$

$$a_6 = -y_4 \quad b_6 = y_3 \quad c_6 = -y_{34} \quad (64)$$

$$a_7 = y_4 \quad b_7 = -y_4 \quad c_7 = y_{34} \quad (65)$$

$$a_8 = 0 \quad b_8 = 0 \quad c_8 = 0 \quad (66)$$

$$a_9 = x_{14} \quad b_9 = -x_{14} \quad c_9 = -x_{34} \quad (67)$$

$$a_{10} = 0 \quad b_{10} = 0 \quad c_{10} = 0 \quad (68)$$

$$a_{11} = x_{14} \quad b_{11} = -x_{14} \quad c_{11} = -x_{34} \quad (69)$$

$$a_{12} = y_4 \quad b_{12} = -y_4 \quad c_{12} = y_{34} \quad (70)$$

$$a_{13} = 0 \quad b_{13} = y_4 \quad c_{13} = 0 \quad (71)$$

$$a_{14} = 0 \quad b_{14} = 0 \quad c_{14} = 0 \quad (72)$$

$$a_{15} = 0 \quad b_{15} = x_{14} \quad c_{15} = -x_{12} \quad (73)$$

$$a_{16} = 0 \quad b_{16} = 0 \quad c_{16} = 0 \quad (74)$$

$$a_{17} = 0 \quad b_{17} = x_{14} \quad c_{17} = -x_{12} \quad (75)$$

$$a_{18} = 0 \quad b_{18} = y_4 \quad c_{18} = 0 \quad (76)$$

$$a_{19} = 0 \quad b_{19} = -y_3 \quad c_{19} = 0 \quad (77)$$

$$a_{20} = 0 \quad b_{20} = 0 \quad c_{20} = 0 \quad (78)$$

$$a_{21} = -x_{12} \quad b_{21} = -x_{23} \quad c_{21} = x_{12} \quad (79)$$

$$a_{22} = 0 \quad b_{22} = 0 \quad c_{22} = 0 \quad (80)$$

$$a_{23} = -x_{12} \quad b_{23} = -x_{23} \quad c_{23} = x_{12} \quad (81)$$

$$a_{24} = 0 \quad b_{24} = -y_3 \quad c_{24} = 0 \quad (82)$$



# STRUCTURAL ELEMENT DESCRIPTIONS

The approach used to evaluate the stiffness matrix [U] is to identify each term in the triple product (equation 56) prior to the integration. Toward this end, the 8 x 8 matrix [U] is thought of as being constructed of sixteen 2 x 2 submatrices, each identified by subscripts (i,j) as in the following table

|       | j = 1 | j = 2 | j = 3 | j = 4 |
|-------|-------|-------|-------|-------|
| i = 1 |       |       |       |       |
| i = 2 |       |       |       |       |
| i = 3 |       |       |       |       |
| i = 4 |       |       |       |       |

2 x 2 Partition  
(Typical)

The subscript i indicates a horizontal slice of 2 rows of the matrix and the subscript j indicates a vertical slice of 2 columns of the matrix. Further if the row identifier, k, and column identifier, l, are introduced to locate an element within any particular 2 x 2 submatrix (k and l assume values of 1 and 2), then a general term in the U matrix is located by specification of i, j, k, and l. Thus, when advantage is taken of some zero terms in the matrix A,

$$U_{ij_{kl}} = h \int_0^1 \int_0^1 \frac{d + en + f\xi + h^*n\xi + pn^2 + q\xi^2}{J} d\xi dn \quad (83)$$

$$d = G_{kl}a_{k_1}a_{l_1} + G_{k3}a_{k_1}a_{l_2} + G_{3l}a_{k_2}a_{l_1} + G_{33}a_{k_2}a_{l_2} \quad (84)$$

$$e = G_{kl}(a_{k_1}b_{l_1} + b_{k_1}a_{l_1}) + G_{k3}(a_{k_1}b_{l_2} + b_{k_1}a_{l_2}) + G_{3l}(a_{k_2}b_{l_1} + b_{k_2}a_{l_1}) + G_{33}(a_{k_2}b_{l_2} + b_{k_2}a_{l_2}) \quad (85)$$

$$f = G_{kl}(a_{k_1}c_{l_1} + c_{k_1}a_{l_1}) + G_{k3}(a_{k_1}c_{l_2} + c_{k_1}a_{l_2}) + G_{3l}(a_{k_2}c_{l_1} + c_{k_2}a_{l_1}) + G_{33}(a_{k_2}c_{l_2} + c_{k_2}a_{l_2}) \quad (86)$$



# MODULE FUNCTIONAL DESCRIPTIONS

$$h^* = G_{k\ell}(b_{k_1}c_{\ell_1} + c_{k_1}b_{\ell_1}) + G_{k3}(b_{k_1}c_{\ell_2} + c_{k_1}b_{\ell_2}) + G_{3\ell}(b_{k_2}c_{\ell_1} + c_{k_2}b_{\ell_1}) + G_{33}(b_{k_2}c_{\ell_2} + c_{k_2}b_{\ell_2}) \quad (87)$$

$$p = G_{k\ell}b_{k_1}b_{\ell_1} + G_{k3}b_{k_1}b_{\ell_2} + G_{3\ell}b_{k_2}b_{\ell_1} + G_{33}b_{k_2}b_{\ell_2} \quad (88)$$

$$q = G_{k\ell}c_{k_1}c_{\ell_1} + G_{k3}c_{k_1}c_{\ell_2} + G_{3\ell}c_{k_2}c_{\ell_1} + G_{33}c_{k_2}c_{\ell_2} \quad (89)$$

where additional subscripts are identified as

$$k_1 = 6(i-1) + 4(k-1) + 1 \quad (90)$$

$$k_2 = 6(i-1) + 3(k-1) + 3 \quad (91)$$

$$\ell_1 = 6(j-1) + 4(\ell-1) + 1 \quad (92)$$

$$\ell_2 = 6(j-1) + 3(\ell-1) + 3 \quad (93)$$

## 3. Gaussian Quadrature

The evaluation of equation (83) is carried out by Gaussian quadrature utilizing a 4 x 4 grid. Thus

$$U_{ij_{k\ell}} = \frac{h}{4} \sum_{m=0}^3 \sum_{n=0}^3 \bar{a}_m \bar{a}_n \left[ \frac{d + e\eta_n + f\xi_m + h^*\eta_n\xi_m + p\eta_n\eta_n + q\xi_m\xi_m}{j(\eta_n, \xi_m)} \right] \quad (94)$$

where

$$\xi_0 = \eta_0 = -\frac{1}{2} \sqrt{\left(\frac{30 + \sqrt{480}}{70}\right)} + \frac{1}{2} = .069431844 \quad (95)$$

$$\xi_1 = \eta_1 = -\frac{1}{2} \sqrt{\left(\frac{30 - \sqrt{480}}{70}\right)} + \frac{1}{2} = .330009478 \quad (96)$$

# STRUCTURAL ELEMENT DESCRIPTIONS

$$\xi_2 = \eta_2 = \frac{1}{2} \sqrt{\left(\frac{30 - \sqrt{480}}{70}\right)} + \frac{1}{2} = .669990522 \quad (97)$$

$$\xi_3 = \eta_3 = \frac{1}{2} \sqrt{\left(\frac{30 + \sqrt{480}}{70}\right)} + \frac{1}{2} = .930568156 \quad (98)$$

$$\bar{a}_0 = \bar{a}_3 = .347854845 \quad (99)$$

$$\bar{a}_1 = \bar{a}_2 = .652145155 \quad (100)$$

4. The 3 x 3 stiffness matrix partition  $[k_{ij}]$  in global coordinates is then given by

$$[k_{ij}] = [T_i]^T ([E][B][U][B]^T[E]^T)_{ij} [T_j] \quad (101)$$

where  $i$  corresponds to the current pivot number and assumes values 1, 2, 3, 4;  
 $j$  corresponds to one of the connected grid points and assumes values 1, 2, 3, 4;  
 $([E][B][U][B]^T[E]^T)_{ij}$  is the appropriate 3 x 3 partition of the 12 x 12 matrix  $[E][B][U][B]^T[E]^T$ .

5. For use in the overall structural matrix, the matrices are expanded to 6 x 6 to form:

$$[K_{ij}] = \begin{bmatrix} k_{ij} & \vdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \vdots & 0 \end{bmatrix} \quad (102)$$

## 4.87.19.4 Mass Matrix Generation for the QDMEM1 (Subroutine MASSTQ of Module SMA2)

The mass matrix for the isoparametric element is calculated in the same manner as QDMEM element which subdivides the quadrilateral into two triangles. See Section 4.87.4.11 on pages 4.87-74 to 4.87-75 of the Programmer's Manual.

## 4.87.19.5 Element Load Calculations for the QDMEM1 Element (Subroutine QDMEM1 of Module SSG1)

The first step in the calculation of the thermal load vector is to compute the following matrix



# MODULE FUNCTIONAL DESCRIPTIONS

$$[C]_{8 \times 3} = h \int_0^1 \int_0^1 [A]^T J d\xi d\eta \quad (103)$$

The integrations in equation (103) can be performed in closed form and results in the following

$$C_{11} = -\frac{hy_4}{2} \quad (104)$$

$$C_{12} = 0 \quad (105)$$

$$C_{13} = -\frac{hx_{24}}{2} \quad (106)$$

$$C_{21} = 0 \quad (107)$$

$$C_{22} = -\frac{hx_{24}}{2} \quad (108)$$

$$C_{23} = -\frac{hy_4}{2} \quad (109)$$

$$C_{31} = \frac{hy_3}{2} \quad (110)$$

$$C_{32} = 0 \quad (111)$$

$$C_{33} = \frac{hx_{13}}{2} \quad (112)$$

$$C_{41} = 0 \quad (113)$$

$$C_{42} = \frac{hx_{13}}{2} \quad (114)$$

$$C_{43} = \frac{hy_3}{2} \quad (115)$$

$$C_{51} = \frac{hy_4}{2} \quad (116)$$



# STRUCTURAL ELEMENT DESCRIPTIONS

$$C_{52} = 0 \quad (117)$$

$$C_{53} = \frac{hx_{24}}{2} \quad (118)$$

$$C_{61} = 0 \quad (119)$$

$$C_{62} = \frac{hx_{24}}{2} \quad (120)$$

$$C_{63} = \frac{hy_4}{2} \quad (121)$$

$$C_{71} = -\frac{hy_3}{2} \quad (122)$$

$$C_{72} = 0 \quad (123)$$

$$C_{73} = -\frac{hx_{13}}{2} \quad (124)$$

$$C_{81} = 0 \quad (125)$$

$$C_{82} = -\frac{hx_{13}}{2} \quad (126)$$

$$C_{83} = -\frac{hy_3}{2} \quad (127)$$

Given the temperature of the element,  $\bar{t}$ , either directly or from the temperature of the four grid points  $t_1$ ,  $t_2$ ,  $t_3$ , and  $t_4$  in the GPTT data block, the routine generates force vectors by the equation:

$$\{P\}_i = [T_i]^T ([E][B][C][G]\{\alpha\})_i (\bar{t} - t_0) \quad (128)$$

where  $([E][B][C][G]\{\alpha\})_i$  is the appropriate  $3 \times 1$  subvector of the  $12 \times 1$  vector  $[E][B][C][G]\{\alpha\}$ ,

$$\{\alpha\} = \begin{Bmatrix} \alpha_x \\ \alpha_y \\ \alpha_{xy} \end{Bmatrix} \quad (129)$$

# MODULE FUNCTIONAL DESCRIPTIONS

and  $t_0$  is the reference or stress-free temperature of the material. The forces are placed in the PG load vector data block.

## 4.87.19.6 Element Stress Calculations for the QDMEM1 Element (Subroutines SQDMM1 and SQDMM2 of Module SDR2)

### 1. Calculations performed in SQDMM1 (Phase 1 calculations).

(a) The stresses in the QDMEM1 elements are calculated at the intersection of the diagonals identified by  $\xi^*$ ,  $\eta^*$ . For parallelograms

$$\xi^* = 1/2 \quad (130)$$

$$\eta^* = 1/2 \quad (131)$$

For other geometries, first  $x^*$  and  $y^*$  must be found as

$$x^* = \frac{y_4 x_{13} x_{12}}{y_3 x_{24} - y_4 x_{13}} \quad (132)$$

$$y^* = - \frac{y_4 (x^* + x_{12})}{x_{24}} \quad (133)$$

provided  $x_{24} \neq 0$ . In the event that  $x_{13} = 0$  a simplification results and

$$x^* = -x_{13} = 0 \quad (134)$$

$$y^* = - \frac{y_4}{x_{24}} x_{12} \quad (135)$$

For the situation where  $x_{24} = 0$

$$x^* = -x_{12} \quad (136)$$

$$y^* = \frac{y_3 x_{12}}{x_{13}} \quad (137)$$



# STRUCTURAL ELEMENT DESCRIPTIONS

After the values of  $x^*$  and  $y^*$  are obtained, in the situation where  $x_{34} \neq -x_{12}$ ,

$$\eta^* = - \frac{b^* \pm \sqrt{(b^*)^2 - 4a^*x_{12}y^*}}{2a^*} \quad (138)$$

where the  $\pm$  sign is chosen such that  $0 < \eta^* < 1$  and

$$b^* = -y_4x_{12} + c^* \quad (139)$$

$$a^* = -y_4x_{23} + y_3x_{14} \quad (140)$$

$$c^* = y_{34}x^* - y^*(x_{34} + x_{12}) \quad (141)$$

Should  $a^* = 0$ , then

$$\eta^* = \frac{-x_{12}y^*}{b^*} \quad (142)$$

Further, if  $y_{34} \neq 0$ ,

$$\xi^* = \frac{-c^* + (y_4x_{23} - y_3x_{14})\eta^*}{y_{34}x_{12}} \quad (143)$$

If  $y_{34} = 0$ ,

$$\xi^* = \frac{x^* + x_{14}\eta^*}{[\eta^*(x_{34} + x_{12}) - x_{12}]} \quad (144)$$

For the situation where  $x_{34} = -x_{12}$  but  $y_{34} \neq 0$

$$\eta^* = \frac{-\hat{b} \pm \sqrt{(\hat{b})^2 + 4\hat{a}x_{12}y^*}}{2\hat{a}} \quad (145)$$

where the  $\pm$  sign is chosen such that  $0 < \eta^* < 1$  and

$$\hat{b} = x_{12}y_4 - y_{34}x^* \quad (146)$$

$$\hat{a} = -x_{14}y_{34} \quad (147)$$



# MODULE FUNCTIONAL DESCRIPTIONS

Should  $\hat{a} = 0$ , then

$$\eta^* = \frac{x_{12}y^*}{\hat{b}} \quad (148)$$

After  $\xi^*$  is evaluated, then

$$\xi^* = \frac{y^* - y_4\eta^*}{y_{34}\eta^*} \quad (149)$$

(b) The values of  $\xi^*$  and  $\eta^*$  are substituted into matrix  $[A]$  and the transformations from displacements to stress are then given by

$$[S_i] = ([G][A][B]^T[E]^T)_i [T_i] \quad i = 1, 2, 3, 4 \quad (150)$$

where  $([G][A][B]^T[E]^T)_i$  is the appropriate  $3 \times 3$  partition of the  $3 \times 12$  matrix  $[G][A][B]^T[E]^T$ . The temperature-stress relation is

$$\{S_t\} = -[G]\{\alpha\} \quad (151)$$

## 2. Calculations performed by SQDMM2 (Phase 2 calculations)

The equation for stress is

$$\begin{pmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{pmatrix} = \sum_{i=1}^4 [S_i]\{u_{g_i}\} + \{S_t\}(\bar{t} - t_0) \quad (152)$$

The principal stresses are:

$$\sigma_1 = \frac{\sigma_x + \sigma_y}{2} + \sqrt{\left(\frac{\sigma_x - \sigma_y}{2}\right)^2 + \sigma_{xy}^2} \quad (153)$$

$$\sigma_2 = \frac{\sigma_x + \sigma_y}{2} - \sqrt{\left(\frac{\sigma_x - \sigma_y}{2}\right)^2 + \sigma_{xy}^2} \quad (154)$$

$$\phi_1 = \frac{1}{2} \tan^{-1} \left( \frac{2\sigma_{xy}}{\sigma_x - \sigma_y} \right) \text{ (converted to degrees)} \quad (155)$$

where  $\phi_1$  is limited to  $-90^\circ \leq \phi_1 \leq 90^\circ$ .

## STRUCTURAL ELEMENT DESCRIPTIONS

The maximum shear stress  $\tau$  is given by

$$\tau = \sqrt{\left(\frac{\sigma_x - \sigma_y}{2}\right)^2 + \sigma_{xy}^2} \quad (156)$$

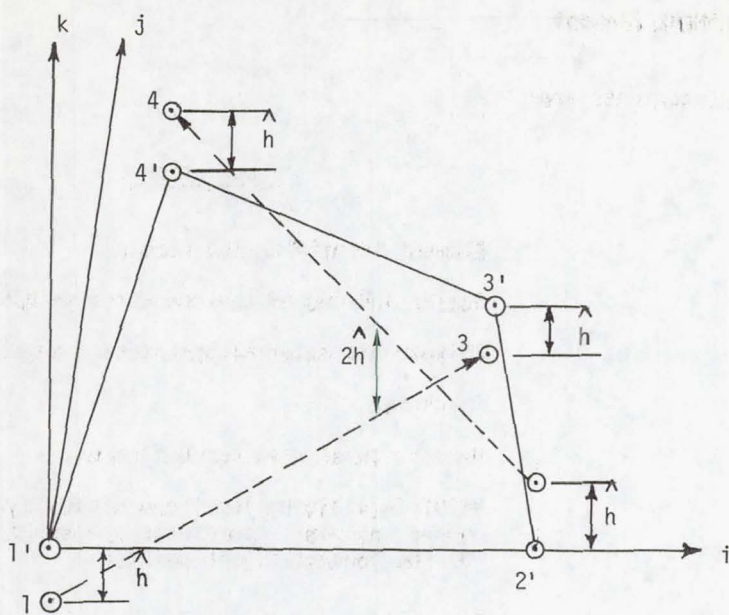


Figure 1. Mean plane for quadrilateral membrane element. (Actual grid points are indicated by unprimed numbers and projection of grid points onto mean plane are indicated by primed numbers.)

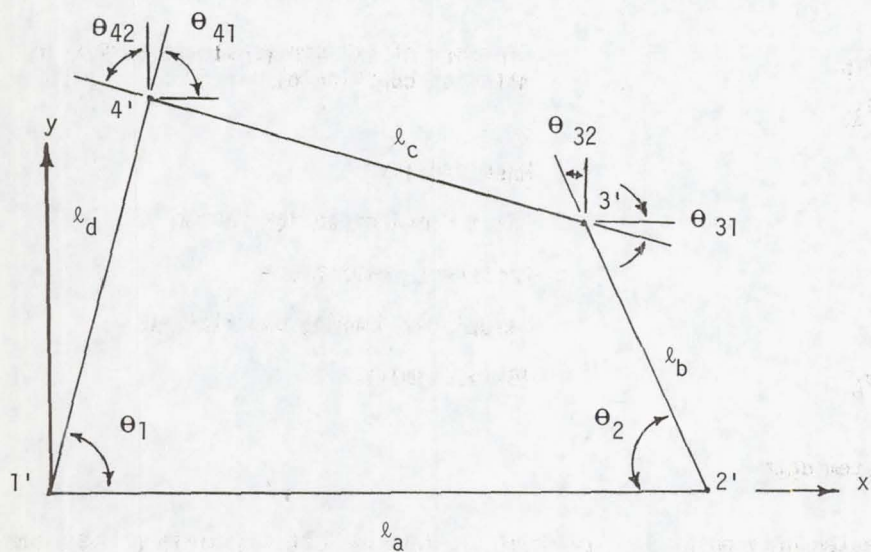


Figure 2. Characteristic lengths and angles for quadrilateral membrane element in mean plane.



## STRUCTURAL ELEMENT DESCRIPTIONS

### 4.87.20 The QDMEM2 Element

#### 4.87.20.1 Input Data for the QDMEM2 Element

1. The ECPT/EST data block contents are:

| <u>Symbol</u>                          | <u>Description</u>   |
|--|--|
| EID                                    | Element identification number  |
| $SIL_i, i = 1,2,3,4$                   | Scalar indices of the connected grid points  |
| $\theta$                               | Anisotropic material orientation angle   |
| $t$                                    | Thickness  |
| $\mu$                                  | Nonstructural mass per unit area   |
| $N_i, X_i, Y_i, Z_i,$<br>$i = 1,2,3,4$ | BGPDT data giving local coordinate system number and basic coordinate system locations for the connected grid points |
| $T_m$                                  | Temperature for material properties  |

2. Material data:

The material subroutine, MAT, returns the following data

| <u>Symbol</u>  | <u>Description</u>   |
|--|--|
| $G_{11}, G_{12}, G_{13}$<br>$G_{22}, G_{23}, G_{33}$ | Elements of 3x3 stress-strain matrix in material coordinates |
| $\rho$   | Mass density   |
| $\{\alpha\}$   | 3x1 thermal expansion vector                                 |
| $T_o$  | Reference temperature  |
| $G_e$  | Structural damping coefficient                               |
| $\sigma_t, \sigma_c, \sigma_s$                       | Stress limits  |

3. Coordinate system data

For each connected grid point,  $i$ , the BGPDT data is used to calculate a 3x3 global-to-basic transformation matrix,  $[T_i]$ .

# MODULE FUNCTIONAL DESCRIPTIONS

## 4.87.20.2 Basic Calculations for the QDMEM2 Element (Subroutine Q2BCS)

1. The element coordinate system is defined by the following equations:

$$\{D_{13}\} = \begin{pmatrix} x_3 - x_1 \\ y_3 - y_1 \\ z_3 - z_1 \end{pmatrix}, \quad (1)$$

$$\{D_{24}\} = \begin{pmatrix} x_4 - x_2 \\ y_4 - y_2 \\ z_4 - z_2 \end{pmatrix}, \quad (2)$$

$$\{N\} = \{D_{13}\} \times \{D_{24}\}, \quad (3)$$

$$A = \frac{1}{2} |\{N\}|, \quad (4)$$

$$\{k\} = \frac{1}{|\{N\}|} \{N\}, \quad (5)$$

$$\{D_{12}\} = \begin{pmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{pmatrix}, \quad (6)$$

$$h = \frac{1}{2} \{D_{12}\} \cdot \{k\}, \quad (7)$$

$$\{x\} = \{D_{12}\} - 2h\{k\}, \quad (8)$$

$$\{i\} = \frac{1}{|\{x\}|} \{x\}, \quad (9)$$

$$\{j\} = \{k\} \times \{i\}, \quad (10)$$



## STRUCTURAL ELEMENT DESCRIPTIONS

The unit vectors are stored in the transformation matrix E as follows:

$$[E] = \begin{bmatrix} \{i\}^T \\ \{j\}^T \\ \{k\}^T \end{bmatrix} . \quad (11)$$

If  $h^2/A < \epsilon_p$ , a flag is set indicating the element lies approximately in a plane.

2. The locations of the points in element coordinates are:

$$\{r_1\} = \begin{pmatrix} 0 \\ 0 \\ -h \end{pmatrix} , \quad (12)$$

$$\{r_i\} = [E] \begin{pmatrix} x_i - x_1 \\ y_i - y_1 \\ z_i - z_1 \end{pmatrix} + \{r_1\} , \quad (13)$$

$$i = 2, 3, 4, 5,$$

where

$$\begin{pmatrix} x_5 \\ y_5 \\ z_5 \end{pmatrix} = \frac{1}{4} [\{x_1\} + \{x_2\} + \{x_3\} + \{x_4\}] . \quad (14)$$

### 4.87.20.3 Subtriangle Calculations for the QDMEM2 Element (Subroutine Q2TRMS)

1. Input data:

| <u>Stiffness</u>                      | <u>Thermal Load</u>                   | <u>Stress Recovery</u>                |
|---------------------------------------|---------------------------------------|---------------------------------------|
| a,b                                   | a,b                                   | a,b                                   |
| [r]                                   | [r]                                   | [r]                                   |
| [G <sub>e</sub> ]                     | [G <sub>e</sub> ]                     | [G <sub>e</sub> ]                     |
| cosθ <sub>m</sub> , sinθ <sub>m</sub> | cosθ <sub>m</sub> , sinθ <sub>m</sub> | cosθ <sub>m</sub> , sinθ <sub>m</sub> |
| t                                     | t                                     | t                                     |
|                                       | {α <sub>e</sub> }(T-T <sub>0</sub> )  | {α <sub>e</sub> }                     |



# MODULE FUNCTIONAL DESCRIPTIONS

where

- $[r]$  is a 5x3 matrix, each row is the location of a point in element coordinates.
- $a, b$  are pointers to the two connected points in the  $[r]$  matrix. The fifth row is always the third point.
- $[G_e]$  is the 3x3 material matrix in material coordinates defined in /MATOUT/.
- $\cos\theta_m, \sin\theta_m$  are the orientation angle cosine and sine.
- $t$  is the element thickness.
- $\alpha_e$  is the thermal expansion coefficient in material coordinates, given in /MATOUT/.
- $(T-T_0)$  is the temperature of the triangle.

## 2. Output data:

| <u>Stiffness</u> | <u>Thermal Load</u> | <u>Stress Recovery</u> |             |
|------------------|---------------------|------------------------|-------------|
| $[K_{ca}]$       | $[K_{ca}]$          | $[K_{ca}]$             | $\{p_a^t\}$ |
| $[K_{cb}]$       | $[K_{cb}]$          | $[K_{cb}]$             | $\{p_b^t\}$ |
| $[K_{cc}]$       | $[K_{cc}]$          | $[K_{cc}]$             | $\{p_c^t\}$ |
| $[K_{aa}]$       | $\{P_a\}$           | $[S_a]$                | $\{Z_a\}$   |
| $[K_{ab}]$       | $\{P_b\}$           | $[S_b]$                | $\{Z_b\}$   |
| $[K_{ba}]$       | $\{P_c\}$           | $[S_c]$                | $\{Z_c\}$   |
| $[K_{bb}]$       |                     | $[K_{aa}]$             |             |
|                  |                     | $[K_{ab}]$             |             |
|                  |                     | $[K_{ba}]$             |             |
|                  |                     | $[K_{bb}]$             |             |

## 3. The equations to produce this data are:

$$\{V_{12}\} = \{r_b\} - \{r_a\},$$

$$\{V_{13}\} = \{r_c\} - \{r_b\},$$

# STRUCTURAL ELEMENT DESCRIPTIONS

$$\{N\} = \{V_{12}\} \times \{V_{13}\},$$

$$\{k\} = \frac{1}{|\{N\}|} \{N\},$$

$$A = \frac{1}{2} |\{N\}|,$$

$$\{i\} = \frac{1}{|\{V_{12}\}|} \{V_{12}\},$$

$$\{j\} = \{k\} \times \{i\},$$

$$[E_2] = \begin{bmatrix} \{i\}^T \\ \{j\}^T \end{bmatrix},$$

$$x_b = |\{V_{12}\}|,$$

$$x_c = \{i\}^T \{V_{13}\},$$

$$y_c = \{j\}^T \{V_{13}\},$$

$$c_1 = \frac{1}{x_b} c_2 = \frac{x_c}{x_b} c_3 = \frac{1}{y_c} c_4 = c_3(c_2 - 1)$$

$$[C_a] = \begin{bmatrix} -c_1 & 0 \\ 0 & c_4 \\ c_4 & -c_1 \end{bmatrix}$$

$$[C_b] = \begin{bmatrix} c_1 & 0 \\ 0 & -c_2 c_3 \\ -c_2 c_3 & c_1 \end{bmatrix}$$

$$[C_c] = \begin{bmatrix} 0 & 0 \\ 0 & c_3 \\ c_3 & 0 \end{bmatrix}$$

# MODULE FUNCTIONAL DESCRIPTIONS

The material orientation cosine ( $C_m$ ) and sine ( $S_m$ ) are calculated by the equations

$$\begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = \{r_b - r_a\},$$

$$l = \sqrt{\Delta x^2 + \Delta y^2},$$

$$C_m = (\cos\theta_m \Delta x + \sin\theta_m \Delta y) / l,$$

$$S_m = (\sin\theta_m \Delta x - \cos\theta_m \Delta y) / l.$$

The material orientation is used to transform anisotropic materials,

$$[T^m] = \begin{bmatrix} C_m^2 & S_m^2 & C_m S_m \\ S_m^2 & C_m^2 & -C_m S_m \\ -2C_m S_m & 2C_m S_m & (C_m^2 - S_m^2) \end{bmatrix}.$$

The equation for each stiffness matrix partition is

$$[K_{ij}] = A_t [H_i^T] [G_e] [H_j],$$

where

$$[H_i] = [T^m] [C_i] [E_2],$$

$$i = a, b, c.$$

The equation for the thermal loads is

$$\{P_i\} = A_t [H_i]^T [G_e] \{\alpha_e\} (T - T_0),$$

$$i = a, b, c.$$

The equations for the stress matrices are

$$[S_i] = [G_e] [H_i],$$

$$\{p_i^t\} = A_t [H_i]^T [G_e] \{\alpha_e\},$$

$$\{Z_i\}^T = t \{T_3^m\}^T [S_i],$$



# STRUCTURAL ELEMENT DESCRIPTIONS

where  $\{T_3^m\}$  is the third column of the  $[T^m]$  matrix,

$$i = a, b, c.$$

## 4.87.20.4 Stiffness Matrix Calculations for the QDMEM2 Element (Subroutine KQDM2S of Module SMA1)

1. The material properties in the material coordinates are calculated by calling subroutine MAT with  $\theta = 0$  degrees.
2. The area of core for storing element stiffness matrix partitions is set to zero, and the basic element calculations are done with subroutine Q2BCS, which returns the following data

$\{r_i\}$ ,  $i = 1, \dots, 5$  The locations of the five points in element coordinates.

$[E]$  The 3x3 basic-to-element coordinate system transformation.

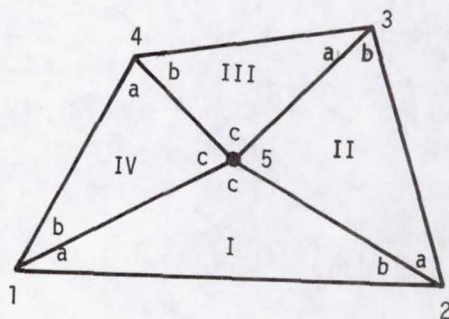
PFLAG Flag indicating whether or not the element approximates a plane.

3. The four subtriangles are processed with subroutine Q2TRMS. Each triangle,  $I$ , is given three points defined by the mapping matrix where

$$\text{Point number} = M_{I\alpha}, \quad \alpha = 1, 2, 3 \\ I = 1, 2, 3, 4,$$

$$[M] = \begin{bmatrix} 1 & 2 & 5 \\ 2 & 3 & 5 \\ 3 & 4 & 5 \\ 4 & 1 & 5 \end{bmatrix}.$$

The orientation of the triangles is shown below



# MODULE FUNCTIONAL DESCRIPTIONS

- Each triangle produces seven matrices,

$$[K_{5a}], [K_{5b}], [K_{5c}], [K_{aa}], [K_{ab}], [K_{ba}], [K_{bb}].$$

These are added to the appropriate matrices for the entire element

$$\begin{aligned} [K_{5i}] \quad i &= 1, \dots, 5, \\ [K_{ij}] \quad i &= 1, \dots, 4, j = 1, \dots, 4. \end{aligned}$$

- The displacement of the center point will be constrained by the equation

$$\{u_5\} = \sum_i [G_i] \{u_i\}.$$

If the element is not planar,

$$G_i = -[K_{55}]^{-1} [K_{5i}]$$

If the element is planar,

$$\bar{K}_i = [K_{5i}],$$

with the third rows modified as follows:

$$[\bar{K}_i] = \begin{bmatrix} x & x & x \\ x & x & x \\ 0 & 0 & -.25 \end{bmatrix} \quad i \neq 5$$

$$[\bar{K}_5] = \begin{bmatrix} x & x & x \\ x & x & x \\ 0 & 0 & 1.0 \end{bmatrix} \quad i = 5$$

The equation for  $G_i$  is

$$[G_i] = -[\bar{K}_5]^{-1} [\bar{K}_i].$$

- The "pivot" point, p, is found and the constrained matrices,  $[K_{ij}^e]$ , are produced.

If the element is not planar,

$$[K_{pj}^e] = [K_{pj}] + [K_{5p}]^T [G_j].$$



## STRUCTURAL ELEMENT DESCRIPTIONS

If the element is planar,

$$[K_{pj}^e] = [K_{pj}] + [K_{5p}]^T [G_j] + [G_p]^T [K_{5j}] + [G_p]^T [K_{55}] [G_j],$$

(p = pivot point, j = 1,2,3,4).

7. The matrices are transformed to global coordinates and output as double precision 6x6 partitions,

$$[K_{pj}^g] = ([E] [T_p])^T [K_{pj}] ([E] [T_j]).$$

Output:

$$[K_{pj}^g] = \begin{bmatrix} K_{pj}^g & & 0 \\ & \text{---} & \\ 0 & & 0 \end{bmatrix}.$$

8. If the structural damping matrix,  $[K_{gg}^h]$ , is being produced, the damping matrices,  $[K_{pj}^h]$  are also output where

$$[K_{pj}^h] = G_e [K_{pj}^0].$$

### 4.87.20.5 Mass Matrix Generation

Subroutine MASSTQ of module SMA2 is used to generate a lumped mass matrix. The element is treated exactly like a QDMEM element.

### 4.87.20.6 Thermal Loads

The basic calculations indicated in Sections 4.87.20.2 and 4.87.20.4 are performed. The average temperature,  $T$ , is given in the GPTT data. The material routine, MAT, is called to produce the  $\{\alpha_e\}$  vector and the reference temperature,  $T_0$ . The following product is passed to the Q2TRMS subroutine,

$$\{d\} = \{\alpha_e\} (T - T_0).$$

The Q2TRMS subroutine will return the three 3x1 load vectors:

$$\{P_a\}, \{P_b\}, \text{ and } \{P_c\}.$$



## MODULE FUNCTIONAL DESCRIPTIONS

It will also return the following 3x3 matrices,

$$[K_{5a}], [K_{5b}], [K_{5c}].$$

The loads are added directly to the five load vectors corresponding to the five connected grid points,  $\{\bar{P}_1\}$ ,  $\{\bar{P}_2\}$ ,  $\{\bar{P}_3\}$ ,  $\{\bar{P}_4\}$ ,  $\{\bar{P}_5\}$ .

The  $[K_{5i}]$  matrices are added to the five 3x3 matrices,

$$[K_{51}], [K_{52}], [K_{53}], [K_{54}], [K_{55}].$$

When all triangles have been processed, the center point load is eliminated by the equation

$$\{P_i^e\} = \{\bar{P}_i\} + [G_i^T] \{\bar{P}_5\} \quad i=1,2,3,4,$$

where the 3x3 matrix,  $[G]$ , is calculated as in step 5, Section 4.87.20.4.

The resulting loads are transformed to global coordinates by the equation

$$\{P_i^g\} = [T]^T [E]^T \{P_i^e\}.$$

4.87.20.7 QDMEM2 Element Stress and Force Calculations (Subroutines SQDM21 and SQDM22 of Module SDR2)

### Phase 1

1. In the Phase 1 calculations, the basic calculations indicated in Sections 4.87.20.2 and 4.87.20.4 are performed, and the material data is extracted with subroutine MAT. The subroutine is called for in each triangle. The output will be

$$[K_{ij}] \quad i = a,b,c \text{ and } j = a,b,c$$

Nine stiffness matrices in element coordinates (3x3).

$$[S_a], [S_b], [S_c]$$

Three stress matrices (3x3).

$$\{P_a^t\}, \{P_b^t\}, \{P_c^t\}$$

Three stress temperature vectors (3x1).

$$\{Z_a\}, \{Z_b\}, \{Z_c\}$$

Three edge force-displacement vectors.

# STRUCTURAL ELEMENT DESCRIPTIONS

2. This data is added to the appropriate positions in the data corresponding to all the connected points of the element. This data is

$$[R_{ij}^e] \quad i = 1,2,3,4 \text{ and } j = 1,2,3,4$$

Sixteen matrices (3x3).

$$[R_{5i}^e] \quad i = 1,2,3,4,5$$

Five matrices (3x3).

$$[S_i^e] \quad i = 1,2,3,4,5$$

Five matrices (3x3).

$$\{P_i^t\} \quad i = 1,2,3,4,5$$

Five vectors (3x1).

3. The  $\{Z_a\}$ ,  $\{Z_b\}$ , and  $\{Z_c\}$  vectors are stored in five 4x3  $[R]$  matrices. For triangle No. I, the data vector,  $\{Z_a\}$ , is stored in row I of the matrix corresponding to point a.
4. The constraints on the center point are applied with the following logic. The matrix,  $[G]$ , is calculated as in step 5 of Section 4.87.20.4. The equations for the reduced data are as follows:

If the planar flag is "off,"

$$[K_{ij}^e] = [R_{ij}^e] + [R_{5i}^e]^T [G_j].$$

If the planar flag is "on,"

$$[K_{ij}^e] = [R_{ij}^e] + [R_{5i}^e]^T [G_j] + [G_i]^T [R_{5j}^e] + [G_i]^T [R_{55}^e] [G_j].$$

For either case,

$$[S_i^e] = \frac{1}{2} [S_i^e] + [S_5^e] [G_i],$$

$$[R_i] = [R_i] + [R_5] [G_i],$$

$$\{P_i^t\} = \{P_i^t\} + [G_i]^T \{P_5^t\}.$$



5. The data is transformed to global coordinates by the equations

$$[K_{ij}^g] = [Q_i] [K_{ij}^e] [E] [T_j],$$

$$\{P_i^g\} = [Q_i] \{P_i^t\},$$

$$[S_i^g] = [S_i^e] [E] [T_i],$$

$$[R_i^g] = [R_i] [E] [T_i],$$

where the  $[Q]$  matrices necessary to transform corner forces to directions parallel to the sides are calculated as follows:

$$\{d_1\} = \{r_2\} - \{r_1\}$$

$$\{d_2\} = \{r_3\} - \{r_2\}$$

$$\{d_3\} = \{r_4\} - \{r_3\}$$

$$\{d_4\} = \{r_1\} - \{r_4\}$$

$$\{k_1\} = -\{d_1\} \times \{d_4\}$$

$$\{k_2\} = -\{d_2\} \times \{d_1\}$$

$$\{k_3\} = -\{d_3\} \times \{d_2\}$$

$$\{k_4\} = -\{d_4\} \times \{d_3\}$$

$$\{\alpha_i\} = \frac{1}{|\{d_i\}|} \{d_i\},$$

$$\{\delta_i\} = \frac{1}{|\{k_i\}|} \{k_i\},$$

$$[Q_i] = [-\{\alpha_i\} \quad \{\alpha_{i-1}\} \quad \{\delta_i\}]^{-1},$$

$$i = 1, 2, 3, 4, \quad (\{\alpha_0\} = \{\alpha_4\}).$$



# STRUCTURAL ELEMENT DESCRIPTIONS

6. The following 250 words of data is saved on the ESTB data block

| <u>Symbol</u>                   | <u>Description</u>  |
|---------------------------------|---|
| ID                              | Element identification  |
| SIL <sub>i</sub><br>i=1,2,3,4   | Connected point SIL values  |
| t                               | Thickness   |
| T <sub>0</sub>                  | Reference temperature   |
| [K <sub>ij</sub> <sup>g</sup> ] | Sixteen point force versus displacement matrices (3x3)              |
| [S <sub>i</sub> <sup>g</sup> ]  | Four stress matrices (3x3)  |
| {P <sub>i</sub> <sup>t</sup> }  | Four point force versus temperature vectors                         |
| {S <sub>t</sub> }               | One stress temperature vector = [G <sub>e</sub> ] {α <sub>e</sub> } |
| [R <sub>i</sub> ]               | Four side force matrices (4x3)                                      |

## Phase 2

The displacement vectors for the connected grid points, {u<sub>i</sub>}, are extracted from open core. The temperature of the element,  $\bar{T}$ , is given in the GPTT data block. If "STRESS" for the element is requested, the following data is calculated:

$$\begin{pmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{pmatrix} = \sum_{i=1}^4 [S_i] \{u_i\} - \{S_t\} (\bar{T} - T_0).$$

The stress output is identical to the QDMEM element output.

If "ELFØRCE" output is requested, the following data is calculated:

$$\{F^i\} = \sum_{j=1}^4 [K_{ij}] \{u_j\} - \{P_i^t\} (T - T_0).$$

# MODULE FUNCTIONAL DESCRIPTIONS

The in-plane forces are:

$$\begin{array}{llll} f_1 = F_2^1 & f_3 = F_2^2 & f_5 = F_2^3 & f_7 = F_2^4 \\ f_2 = F_1^1 & f_4 = F_1^2 & f_6 = F_1^3 & f_8 = F_1^4 \end{array} ,$$

The "kick" forces are:

$$K_i = F_3^i, \quad i=1,2,3,4.$$

The "shear" forces are:

$$\begin{pmatrix} q_1 \\ -q_2 \\ q_3 \\ -q_4 \end{pmatrix} = \sum [R_i] \{u_i\}.$$



## DETERMINANT METHOD OF EIGENVALUE EXTRACTION

### 4.88 DETERMINANT METHOD OF EIGENVALUE EXTRACTION

NASTRAN contains two double precision versions of the determinant method. One version is for the solution of real eigenvalue problems; and one version is for the solution of complex eigenvalue problems. The specifications for both versions are discussed in this document. Real arithmetic is used for the real problem, complex for the complex problem.

#### 4.88.1 Fundamentals of the Determinant Method

The basic notion employed in the determinant method of eigenvalue extraction is very simple. If the elements in a matrix  $[A]$  are functions of the operator  $p$ , then the determinant of  $[A]$  can be expressed as:

$$D([A]) = (p-p_1) (p-p_2) \dots (p-p_n), \quad (1)$$

where  $p_1, p_2, p_3 \dots p_n$  are the eigenvalues of the matrix. The value of the determinant vanishes for  $p = p_i, i = 1, 2, \dots, n$ .

In the determinant method, the determinant is evaluated for trial values of  $p$ , selected according to some iterative procedure, and a criteria is established to determine when  $D([A])$  is sufficiently small or when  $p$  is sufficiently close to an eigenvalue. The eigenvector is then found by solution of the equation:

$$[A] \{u\} = 0, \quad (2)$$

with one of the elements of  $\{u\}$  set to unity.

#### 4.88.2 Evaluation of Determinant

The most convenient procedure for evaluating the determinant of a matrix employs the triangular decomposition of the matrix:  $[A] = [L] [U]$  where  $[L]$  is a lower unit triangle (unit values on the diagonal). The determinant of  $[A]$  is equal to the product of the diagonal terms of  $[U]$ . The matrix  $[A]$  may be expressed as

$$[A] = -p[M] + [K], \quad (3)$$

for real eigenvalue problems and as



# MODULE FUNCTIONAL DESCRIPTIONS

$$[A] = p^2[M] + p[B] + [K], \quad (4)$$

for complex eigenvalue problems.

## 4.88.3 Iteration Algorithm

Consider a series of determinants  $D_{k-2}^{(i)}$ ,  $D_{k-1}^{(i)}$ ,  $D_k^{(i)}$  evaluated for trial values of the eigenvalue  $p = p_{k-2}$ ,  $p_{k-1}$ ,  $p_k$ . Then a better approximation to the eigenvalue is obtained from the following calculations:

Let

$$h_k = p_k - p_{k-1}, \quad (5)$$

$$\lambda_k = h_k / h_{k-1}, \quad (6)$$

$$\delta_k = 1 + \lambda_k. \quad (7)$$

Then

$$h_{k+1} = \lambda_{k+1} h_k, \quad (8)$$

$$p_{k+1} = p_k + h_{k+1}, \quad (9)$$

where

$$\lambda_{k+1} = \frac{-2 D_k^{(i)} \delta_k}{g_k \pm [g_k^2 - 4 D_k^{(i)} \delta_k \lambda_k (D_{k-2}^{(i)} \lambda_k - D_{k-1}^{(i)} \delta_k + D_k^{(i)})]^{1/2}}, \quad (10)$$

$$g_k = D_{k-2}^{(i)} \lambda_k^2 - D_{k-1}^{(i)} \delta_k^2 + D_k^{(i)} (\lambda_k + \delta_k). \quad (11)$$

The (+) or (-) sign in Equation 10 is selected to minimize the absolute value of  $\lambda_{k+1}$ . In the case where  $p_k$ ,  $p_{k-1}$  and  $p_{k-2}$  are all arbitrarily selected initial values (starting points), the starting points should be arranged such that  $|D_k| \leq |D_{k-1}| \leq |D_{k-2}|$  and the (+) or (-) sign in Equation 10 should be selected to minimize the distance from  $p_{k+1}$  to all 3 starting points.

In a real eigenvalue analysis it is possible to calculate a complex  $\lambda_{k+1}$ . In order to

## DETERMINANT METHOD OF EIGENVALUE EXTRACTION

preclude the use of complex arithmetic in a real eigenvalue analysis problem, only the real part of the  $\lambda_{k+1}$  should be used to estimate a  $p_{k+1}$ .

### 4.88.4 Scaling

In calculating the determinant of  $[A]$ , the determinant is scaled by powers of 10 since the accumulated product will rapidly overflow or underflow the floating point size of a digital computer. All operations using the determinant are calculated in scaled arithmetic.

### 4.88.5 Sweeping of Previously Extracted Eigenvalues

Once an eigenvalue has been found to satisfactory accuracy, a return to that eigenvalue by the iteration algorithm can be prevented by dividing the determinant by the factor  $(p-p_i')$  where  $p_i'$  is the accepted approximation to  $p$  in all subsequent calculations.

Thus:

$$D^{(1)}([A]) = \frac{D([A])}{p-p_1'} \quad (12)$$

should be used in place of  $D([A])$  after the first eigenvalue has been found. In general, the reduced determinant used for finding the  $i+1^{\text{st}}$  eigenvalue is:

$$D^{(i)}([A]) = \frac{D^{(i-1)}([A])}{(p-p_i')} = \frac{D([A])}{(p-p_1')(p-p_2')\cdots(p-p_i')} \quad (13)$$

This sweeping procedure is quite satisfactory provided that all  $p_i'$  have been calculated to an accuracy that is limited only by round-off error.

For problems in which zero is an eigenvalue, the number of such eigenvalues is specified by the user. In using the determinant method, zero eigenvalues should be eliminated from the determinant by a preliminary operation,

$$D^{(0)}([A]) = \frac{D([A])}{p^m}, \quad (14)$$

where  $m$  is the multiplicity of the zero eigenvalue.



## MODULE FUNCTIONAL DESCRIPTIONS

For problems with conjugate complex eigenvalues (complex eigenvalue analysis with real matrices) the conjugates of extracted eigenvalues should also be swept from the determinant. Thus

$$D^{(i)}([A]) = \frac{D^{(i-1)}([A])}{(p-p_i')(p-\bar{p}_i')} , \quad (15)$$

where  $\bar{p}_i'$  is the conjugate of  $p_i'$ . It should be noted that the sweeping equations are indeterminate for  $p = p_i'$ . This situation will occur when a root coincides with a starting point or a new estimate with a root already extracted. When the first situation occurs, the starting point should be moved away from the root. When the second situation occurs,  $D_{k+1}$  should be set equal to  $D_k$ .

### 4.88.6 Convergence Criteria

Convergence criteria are based on successive values of the increment  $h_k$  in the estimated eigenvalue. No tests on the magnitude of the determinant or any of the diagonal terms of the triangular decomposition are necessary or desirable. Wilkinson<sup>(1)</sup> shows that for  $h_k$  sufficiently small, the magnitude of  $h_k$  is approximately squared for each successive iteration when using Muller's method. This is an extremely rapid rate of convergence. In a very few iterations the "zone of indeterminacy" is reached within which  $h_k$  remains small but exhibits random behavior due to round-off error. Wilkinson states that if it is desired to calculate the root to the greatest possible precision, the convergence criterion for accepting  $p_k$  as a root should be:

$$|h_{k+1}| > |h_k|. \quad (16)$$

The determinant method accepts this advice, tempered by practical considerations. The first of these is that Equation 16 may be falsely satisfied during the first few iterations while the root tracking algorithm is picking up the "scent". Thus it must, in addition, be required that  $|h_k|$ ,  $|h_{k-1}|$  and  $|h_{k-2}|$  be reasonably small. The second practical consideration is that we may waste a few iterations within the zone of indeterminacy while waiting for Equation 16 to be satisfied. This is avoided by accepting  $p_k$  if  $|h_k|$  is sufficiently small. Finally, if the number of iterations becomes excessively large without satisfying a convergence criterion, the Determinant Method assumes the existence of one iteration loop, gives up and proceeds to



# DETERMINANT METHOD OF EIGENVALUE EXTRACTION

a new set of starting points.

Figure 1 is a flow diagram of a set of tests which meet the requirements discussed above for real eigenvalue problems. The tests are based on calculated values of  $\bar{H}_1$ ,  $\bar{H}_2$ , and  $\bar{H}_3$  which are defined as:

$$\bar{H}_1 = |h_{k-1}| / \sqrt{|p_k|}, \quad (17)$$

$$\bar{H}_2 = |h_k| / \sqrt{|p_k|}, \quad (18)$$

$$\bar{H}_3 = |h_{k+1}| / \sqrt{|p_k|}, \quad (19)$$

where  $p_k = k^{\text{th}}$  estimate of an eigenvalue and  $h_k = p_k - p_{k-1}$ .

A similar set of tests is described in Figure 2 for complex eigenvalue problems. In this case  $\bar{H}_1$ ,  $\bar{H}_2$  and  $\bar{H}_3$  are defined as:

$$\bar{H}_1 = |h_{k-1}|, \quad (20)$$

$$\bar{H}_2 = |h_k|, \quad (21)$$

$$\bar{H}_3 = |h_{k+1}|. \quad (22)$$

The magnitude of the convergence criterion  $\epsilon$  should be selected as a compromise between running time and accuracy. Currently  $\epsilon = 10^{-11}$ . If failure occurs because the number of iterations exceeds the iteration limit, NIT, for two successive sets of starting points,  $\epsilon$  is increased by a factor of 10. If successive pairs of failures still occur,  $\epsilon$  is again increased until the number of permissible changes in  $\epsilon$  is exceeded. The user is informed of the reduced precision of the calculations.

Since eigenvalues are swept out after they are found, all sets of starting points will eventually lead to failure by the preliminary range checks or through successive iteration failure. When this occurs it is presumed that all eigenvalues within the range of interest have been found and the calculations are halted. If for some reason this does not occur the calculation must still be halted. The one remaining avenue for the computer to continue calculations indefinitely is if it continues to find roots. To block this avenue, the calcula-

tions are stopped if the number of roots found exceeds the maximum number of  $N_{EVM}$ . As a safeguard the order in which the roots are found is indicated to the user.

#### 4.88.7 Extraction of Eigenvectors

Once an approximate eigenvalue  $p_j$  has been accepted, the eigenvector is determined by back substitution into the previously computed triangular decomposition of  $[A(p_j)]$ . Now since

$$[A(p_j)]\{u\} = [L(p_j)][U(p_j)]\{u\} = 0, \quad (23)$$

we work only with  $[U(p_j)]$ . Because partial pivoting (row interchanges) have been used, the last diagonal term in  $[U(p_j)]$  will normally be the only term with a very small value. The normal appearance of  $[U(p_j)]$  is as follows, for  $n = 7$ :

$$\begin{bmatrix} X & & & & 0 & 0 & 0 \\ & X & & & X & 0 & 0 \\ & & X & & X & X & 0 \\ & & & X & X & X & X \\ & 0 & & & X & X & X \\ & & & & & X & X \\ & & & & & & \delta \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \end{Bmatrix} = \{0\}. \quad (24)$$

The terms in the upper right corner are zero due to bandwidth.  $\delta$  is a very small number. The eigenvector may be extracted by assigning an arbitrary value (such as 1.0) to  $u_7$  and solving successively for  $u_6, u_5$  etc., from the preceeding rows. Note that this is equivalent to placing a load vector  $\{F\}$  on the right-hand side that is null except for the last term which is set equal to  $\delta$ .

Situations may occur in which  $U_{nn}$  is not the smallest diagonal term. Let  $U_{ii}$  be the smallest diagonal term with  $i < n$ . The most common reason for this occurrence is that the degrees of freedom  $u_{i+1}, u_{i+2}, \dots, u_n$  are, for some reason, uncoupled to the preceeding degrees of freedom. In this case all of the elements in the  $i^{th}$  row of  $[U(p_j)]$  will be very small as shown for  $i = 4, n = 7$ :



# DETERMINANT METHOD OF EIGENVALUE EXTRACTION

$$\begin{bmatrix}
 X & & & & 0 & 0 & 0 \\
 & X & & & X & 0 & 0 \\
 & & X & & X & X & 0 \\
 & & & \delta_{44} & \delta_{45} & \delta_{46} & \delta_{47} \\
 & 0 & & & X & X & X \\
 & & & & & X & X \\
 & & & & & & X
 \end{bmatrix}
 \begin{Bmatrix}
 u_1 \\
 u_2 \\
 u_3 \\
 u_4 \\
 u_5 \\
 u_6 \\
 u_7
 \end{Bmatrix}
 = \{0\}. \quad (25)$$

In the event of multiple or pathologically close eigenvalues two or more rows of  $[U(p_j)]$  will consist of very small values, exhibited below for the very exceptional case where the  $n^{\text{th}}$  row is not very small:

$$\begin{bmatrix}
 X & & & & 0 & 0 & 0 \\
 & X & & & X & 0 & 0 \\
 & & X & & X & X & 0 \\
 & & & \delta_{44} & \delta_{45} & \delta_{46} & \delta_{47} \\
 & 0 & & & X & X & X \\
 & & & & & \delta_{66} & \delta_{67} \\
 & & & & & & X
 \end{bmatrix}
 \begin{Bmatrix}
 u_1 \\
 u_2 \\
 u_3 \\
 u_4 \\
 u_5 \\
 u_6 \\
 u_7
 \end{Bmatrix}
 = \{0\}. \quad (26)$$

In order to accommodate the exceptional cases described above with the more general case of  $\delta = U_{nn}$ , a full load vector  $\{F\}$  is used for the eigenvector calculations. The load vector will also contain elements of the same order of magnitude as the smallest diagonal element of the triangularized matrix  $[U(p_j)]$  in order to prevent digital overflow when the eigenvector is calculated. In addition, a distinct load vector is formed for each eigenvalue to ensure that independent eigenvectors are calculated for multiple or pathologically close eigenvalue problems. The following equations are used for  $\{F\}$ . For real eigenvalues, we have

$$F_i = \frac{\delta(-1)^{ij}}{1 + (1 - \frac{i}{n})^j} ; \quad (27)$$



# MODULE FUNCTIONAL DESCRIPTIONS

For complex eigenvalues,

$$\text{Re}(F_i) = \frac{|\delta| (-1)^{ij}}{1.0 + (1.0 - \frac{1}{n}) j} , \quad (28)$$

$$\text{Im}(F_i) = 0.0, \quad (29)$$

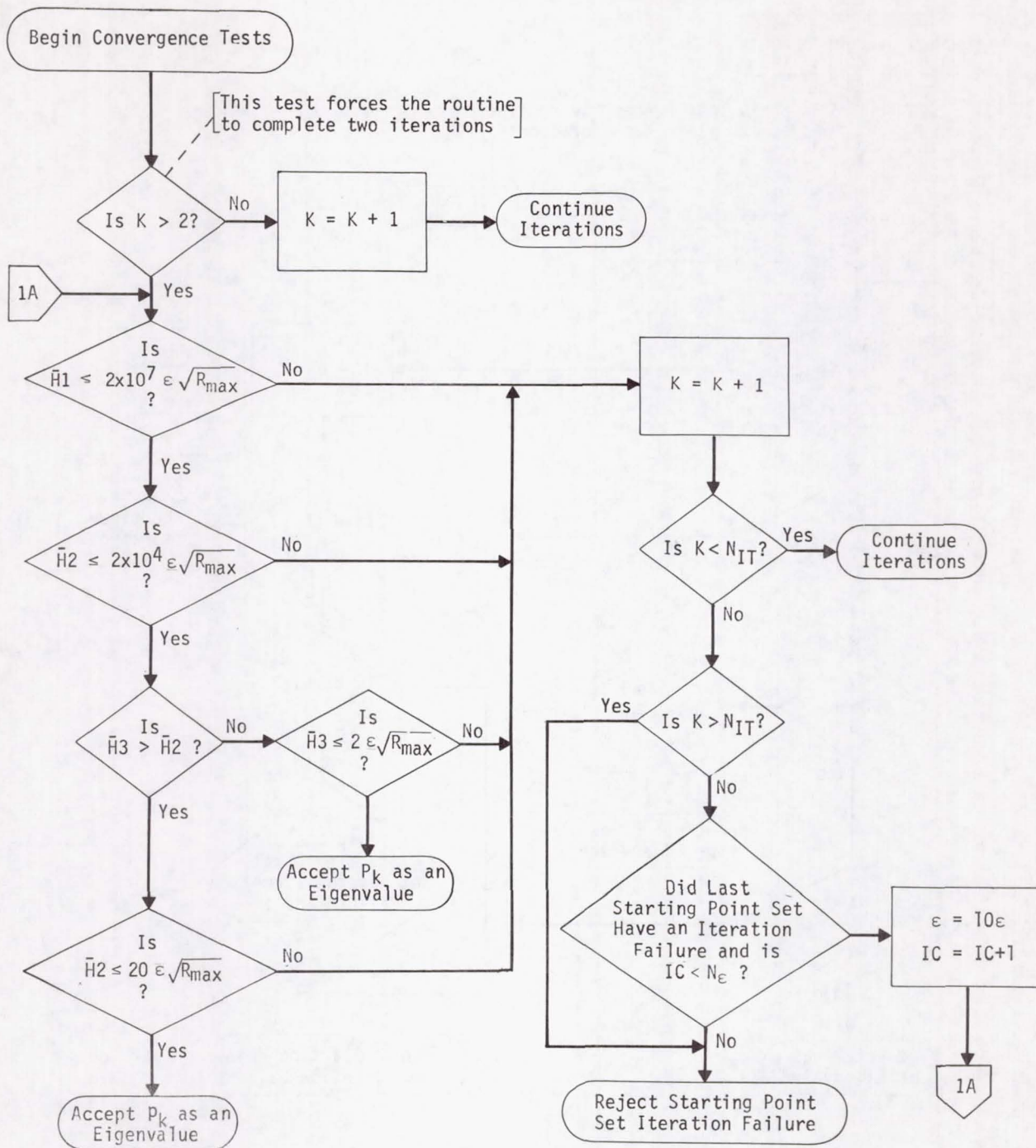
where  $\delta$  = smallest  $U_{ii}$ ,  $j$  = eigenvalue count,  $i = i^{\text{th}}$  element of  $\{F\}$  and  $n$  = number of rows.

There is a possibility that the smallest diagonal element of  $[U]$  may be exactly zero for some eigenvalue. This will be the case when the accepted eigenvalue ( $p_j$ ) is exactly equal to an eigenvalue of the problem. When this occurs,  $\delta = 1.0 \times 10^{-8}$ . The calculated eigenvectors are normalized to a unit maximum real element value.

## REFERENCE

1. Wilkinson, J.H., "The Algebraic Eigenvalue Problem", Clarendon Press, Oxford, 1965.

# DETERMINANT METHOD OF EIGENVALUE EXTRACTION



$\epsilon$  - Convergence Criterion

K - Iteration Counter

IC - Criterion Change Counter

$$\bar{H}_1 = |h_{k-1}| / \sqrt{|p_k|}$$

$$\bar{H}_2 = |h_k| / \sqrt{|p_k|}$$

$$\bar{H}_3 = |h_{k+1}| / \sqrt{|p_k|}$$

Figure 1. Real eigenvalue problems convergence tests

# MODULE FUNCTIONAL DESCRIPTIONS

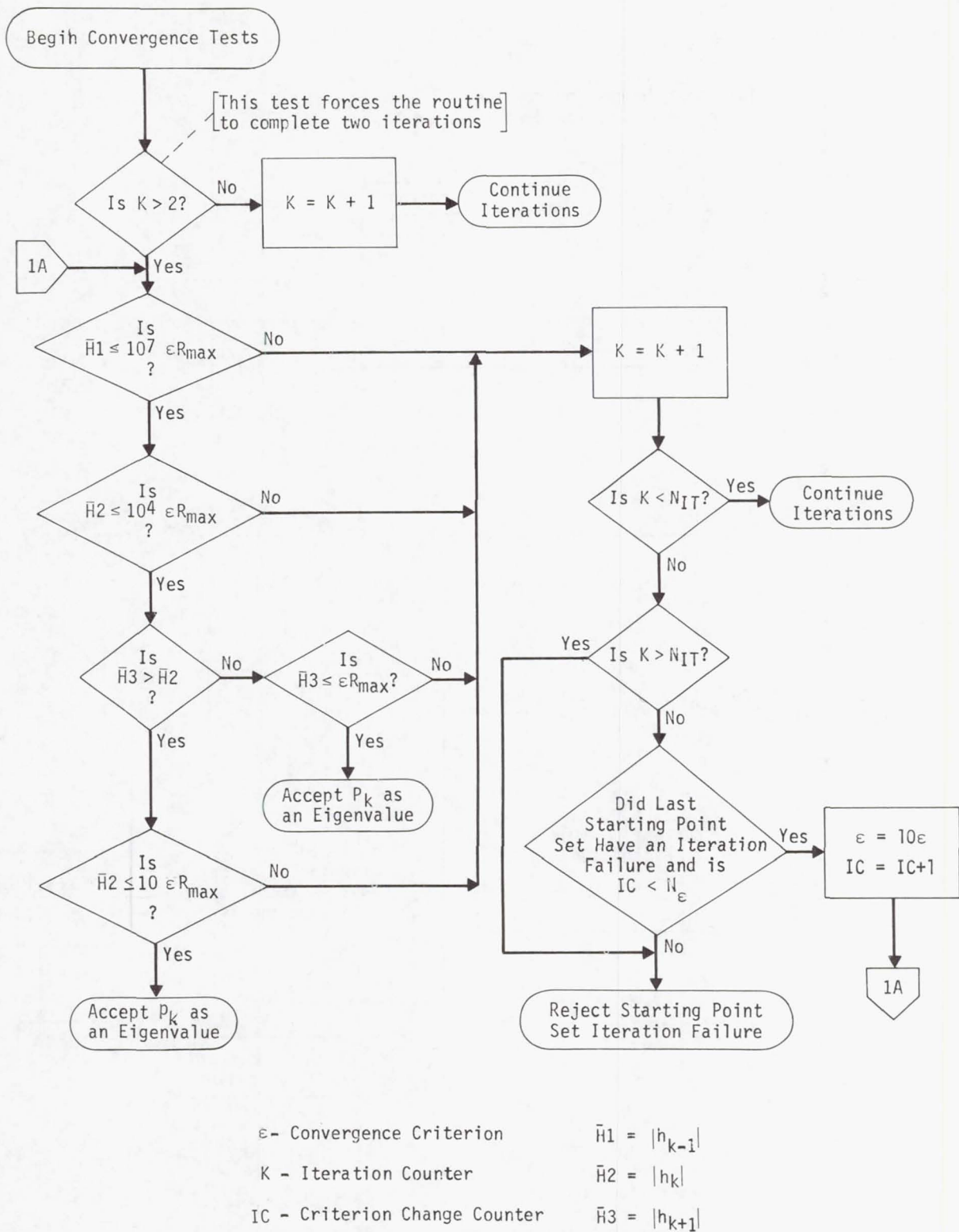


Figure 2. Complex eigenvalue problems convergence tests



## EXECUTIVE PREFACE MODULE IFP4 (INPUT FILE PROCESSOR - PHASE 4)

### 4.89 EXECUTIVE PREFACE MODULE IFP4 (INPUT FILE PROCESSOR - PHASE 4)

#### 4.89.1 Entry Point: IFP4

#### 4.89.2 Purpose

1. To convert the data card images related to fluid and hydroelastic analysis into conventional data card images as output from the IFP module.
2. To calculate data related to the boundaries and the harmonic degrees of freedom of the axisymmetric fluid and append this data to the MATP00L data block.

#### 4.89.3 Calling Sequence

CALL IFP4. IFP4, an Executive Preface Module, is called only by the Preface driver SEMINT.

#### 4.89.4 Input Data Blocks

AXIC - Bulk Data Deck cards related to the hydroelastic problem as output from IFP.

GE0M1 - Grid Point and Coordinate System Data.

GE0M2 - Scalar Point and Element Connection Data.

GE0M4 - Constraint Data.

MATP00L - Direct Input Matrix Data.

#### 4.89.5 Output Data Blocks

Same as the Input Data Blocks.

#### 4.89.6 Parameters

Not applicable to IFP4.

#### 4.89.7 Method

IFP4 allocates the available core as it proceeds. Each type of data card image on the AXIC data block is read and used to form tables or new data card images. The new data and any existing data on the Input data blocks are merged and written on one of two scratch files. After the scratch file data are complete the data are then copied back on the Input/Output data files. (This is not normally allowed. The preface modules, however, have the privilege of writing on an input file.)

# EXECUTIVE PREFACE MODULE IFP4 (INPUT FILE PROCESSOR - PHASE 4)

Data cards as referenced below refer to card images as found on the AXIC input data block. The actual data cards are read and first processed by the IFP. The fluid data card types found on the AXIC data block, and used by IFP4, are listed below along with a list of the output card images produced as a result of their presence, and the data blocks effected.

| <u>IFP4 Input<br/>Card Image</u> | <u>IFP4 Output<br/>Card Image</u> | <u>Data Block<br/>Effected</u> |
|----------------------------------|-----------------------------------|--------------------------------|
| AXIF                             | none                              | all below                      |
| BDYLIST                          | data                              | MATPØØL                        |
| CFLUID2                          | CFLUID2                           | GEØM2                          |
| CFLUID3                          | CFLUID3                           | GEØM2                          |
| CFLUID4                          | CFLUID4                           | GEØM2                          |
| DMIAx                            | DMIG                              | MATPØØL                        |
| FLSYM                            | data (header)                     | MATPØØL                        |
| FREET                            | SPØINT<br>MPC                     | GEØM2<br>GEØM4                 |
| FSLIST                           | CMFREE<br>SPC                     | GEØM2<br>GEØM4                 |
| GRIDB                            | GRID                              | GEØM1                          |
| PRESPT                           | SPØINT<br>MPC                     | GEØM2<br>GEØM4                 |
| RINGFL                           | GRID<br>SEQGP                     | GEØM1<br>GEØM1                 |

It should be noted that the output card images may be a function of several types of input card images as detailed in the following.

4.89.7.1 Data values found on the AXIF card are first stored in core for subsequent use. They are:

1.  $CS_f$ , the coordinate system number for the fluid system
2.  $g$ , the value of gravity
3.  $\rho_d$ , the default value of fluid density
4.  $B_d$ , the default value of the compression coefficient
5.  $NØSYM$ , an integer 0 or 1 indicating whether the unsymmetric (\*series) terms are used in the computations.



## MODULE FUNCTIONAL DESCRIPTIONS

6. A list of harmonic numbers  $n_j$ ,  $j = 1, 2, \dots, J$ , indicating the harmonics to be formulated. If none are supplied by the user, it is implied that the fluid is not to be solved, however the processing of the boundary points (GRIDB) may be necessary as discussed later.

The list of harmonic numbers ( $n_j$ ) are converted to an in core list of index numbers ( $I_j$ ) as follows.

If  $NOSYM=0$ , implying only the symmetric series:

$$I_j = 2n_j + 2, \quad n_j \geq 0, \quad (1)$$

for  $j = 1$  to  $J$ .

If  $NOSYM=1$ , implying the symmetric and unsymmetric (\*) series:

$$I_j = 2n_j + 2, \quad n_j \geq 0, \quad (2)$$

for  $j = 1$  to  $J$ , plus the additional indices:

$$I_j^* = 2n_j + 1, \quad n_j > 0, \quad (3)$$

for  $j = 1$  to  $J$ .

The list of indices as formed above is sorted and held in core for subsequent use. The complete list of indices may thus be up to  $2J$  in length. Henceforth the number of indices in the list is referred to as  $N$ .

### 4.89.7.2 GEOM1 Data Block Processing

1. The GEOM1 file is read and the coordinate system as specified by  $CS_f$  is located among the CORD1C, CORD2C, CORD1S, or CORD2S card images. Its type (cylindrical or spherical) is saved as a flag for use in later processing. If the coordinate system is not located among the above card types a fatal error is indicated to the user and a cylindrical type is assumed to permit further checking of data.
2. GRIDB card images are read and stored in core, 5 words per image. A GRIDB card image defines a normal grid point except that it's location is fixed to a fluid (RINGFL) point.
3. If any GRIDB card images are present IFP4 at this time forms a boundary list table in core.



# EXECUTIVE PREFACE MODULE IFP4 (INPUT FILE PROCESSOR - PHASE 4)

4. For each fluid point,  $IDF_j$ , contained in a BDYLIST card image, a seven word entry is placed in the boundary list table. The contents of this entry are, using data from the BDYLIST card image:

- |    |             |                                 |
|----|-------------|---------------------------------|
| 1. | $IDF_j$     |                                 |
| 2. | 1           | } Integer 1's (temporary flags) |
| 3. | 1           |                                 |
| 4. | 1           |                                 |
| 5. | $IDF_{j-1}$ |                                 |
| 6. | $IDF_{j+1}$ |                                 |
| 7. | $\rho_b$    |                                 |

where  $j$  indicates the respective IDF in the BDYLIST list of IDFs.

If  $IDF_{j-1}$  or  $IDF_{j+1}$  does not exist a zero (0) is entered.

If  $IDF_{j-1}$  or  $IDF_{j+1}$  is designated to be AXIS then a minus one (-1) is entered.

Should  $\rho_b$  as specified in the BDYLIST card image be missing, the default  $\rho_d$ , as specified on the AXIF card image, is used for position 7 of the entry. If both are missing, a user fatal error results. Missing is denoted by an integer -1.

After all BDYLIST card images have been processed and the entries added to the boundary list table, a sort is performed such that the entries are in sort by the primary IDF (found in the first position of each entry).

5. An initial "pass" of RINGFL card images is now made. The meridinal angles ( $x_2$  for a cylindrical coordinate system or  $x_3$  for a spherical system) must be zero and are checked. A binary search is performed to find one or more entries whose primary IDF matches the IDF of each RINGFL card image. When found the values  $X1$ ,  $X2$ , and  $X3$  of the respective images replace the three integer ones in position 2, 3, and 4 of that entry. If an entry is not found, a user fatal error is indicated.

If after all RINGFL card images have been passed, any of the entries in the boundary list table (residing in core) still contain the three integer ones in positions 2, 3, and 4, a user fatal error message is indicated for those particular BDYLIST identification numbers ( $IDF_j$ s).

# MODULE FUNCTIONAL DESCRIPTIONS

6. At this time a normal GRID card image is created from each GRIDB image and merged along with existing GRID card images on GEØM1. Additional GRID card images will be added to GEØM1 in subsequent manipulations.

For each GRIDB card image now residing in core (note 4.89.7.2-2) a normal GRID card is created and consists of the following eight values.

| <u>Field</u> | <u>Symbol</u>   | <u>Description</u>   |
|--------------|-----------------|--|
| 1            | $ID_g$          | ID given on GRIDB image.   |
| 2            | $CS_\ell$       | $CS_f$ from the AXIF image.  |
| 3-5          | $x_1, x_2, x_3$ | <p>These values are formulated by finding <math>X1, X2, X3</math> in the boundary list table entry whose primary identification number matches the reference identification number (IDF) given on the GRIDB card image, and then:</p> <p><math>x_1 = X1</math></p> <p><math>x_2 = \phi</math> if <math>CS_f</math> is cylindrical, or <math>X2</math> if <math>CS_f</math> is spherical.</p> <p><math>x_3 = X3</math> if <math>CS_f</math> is cylindrical, or <math>\phi</math> if <math>CS_f</math> is spherical.</p> <p>Where <math>\phi</math> is supplied by the GRIDB card image.</p> |
| 6            | $CS_d$          | CD from the GRIDB image.   |
| 7            | PS              | PS from the GRIDB image.   |
| 8            | 0               | Not used.  |

The resulting GRID data card images are merged with the existing GRID cards on data blocks GEØM1. If no harmonics exist for the fluid, the module processing is complete.

7. To generate the nonsymmetric connection tables for the boundary, the boundary list table is further altered at this time to result in a table listing the geometry and related grid points for each boundary fluid point.
  - a. For each fluid point entry now in the boundary list table the values  $X1, X2$ , and  $X3$  are converted to  $r$  and  $z$  values which are stored respectively in place of  $X1$  and  $X2$ . If  $CS_f$  is cylindrical then:

EXECUTIVE PREFACE MODULE IFP4 (INPUT FILE PROCESSOR - PHASE 4)

$$\left. \begin{aligned} r &= X1 \\ z &= X3 \\ X2 &\text{ must be zero.} \end{aligned} \right\} \quad (4)$$

If  $CS_f$  is spherical then:

$$\left. \begin{aligned} r &= X1 \sin\left(\frac{\pi}{180} X2\right) \\ z &= X1 \cos\left(\frac{\pi}{180} X2\right) \\ X3 &\text{ must be zero.} \end{aligned} \right\} \quad (5)$$

- b. For each set of three fluid point identification numbers (position 1, 5, and 6 of each entry), the three pairs of coordinates are extracted. The primary values  $r_j$  and  $z_j$  are given with each entry. The secondary values  $r_{j-1}$ ,  $z_{j-1}$ ,  $r_{j+1}$ ,  $z_{j+1}$  must be found by finding the entries which have the same primary identification number as the secondary identification numbers ( $IDF_{j-1}$  or  $IDF_{j+1}$ ) in question. If "axis" (-1) is a secondary identification number, then:

$$\left. \begin{aligned} r_{\text{axis}} &= 0 \\ z_{\text{axis}} &= z_j \end{aligned} \right\} \quad (6)$$

If "not available" (0) is a secondary identification number, then:

$$\left. \begin{aligned} r &= r_j \\ z &= z_j \end{aligned} \right\} \quad (7)$$

The values  $\ell$ ,  $c$ , and  $s$  are calculated and replace the 4th, 5th and 6th word of each entry in the boundary point list at this time such that each entry will now contain:



# MODULE FUNCTIONAL DESCRIPTIONS

| Field | Symbol           | Description   |
|-------|------------------|---|
| 1     | IDF <sub>j</sub> | Fluid point identification                                  |
| 2     | r <sub>j</sub>   | Radial location   |
| 3     | z <sub>j</sub>   | Vertical location   |
| 4     | ℓ                | Length and associated angle components of a conical section |
| 5     | c                |   |
| 6     | s                |   |
| 7     | ρ <sub>b</sub>   | Fluid density   |

where:

$$\ell = \sqrt{\Delta r^2 + \Delta z^2} \quad (8)$$

$$c = \frac{\Delta z}{\ell} \quad (9)$$

$$s = \frac{\Delta r}{\ell} \quad (10)$$

and:

$$\Delta r = \frac{1}{2} \left\{ r_{j+1} - r_{j-1} + \frac{1}{4r_j} [(r_{j+1} - r_j)^2 - (r_{j-1} - r_j)^2] \right\} \quad (11)$$

$$\Delta z = \frac{1}{2} \left\{ z_{j-1} - z_{j+1} + \frac{1}{4r_j} [(r_{j+1} - r_j)(z_j - z_{j+1}) - (r_j - r_{j-1})(z_{j-1} - z_j)] \right\} \quad (12)$$

This list, now referred to as the "boundary point geometry table", remains in core.

The values ℓ, s, and c correspond to the cross section length, and the sine and cosine of the angle ψ as given in Equation 14, Section 16.1.5 of the Theoretical Manual.

- c. As any number of GRIDB points may be connected to a fluid point, the GRIDB card images are now sorted on the referenced fluid point identification (the fifth word of each GRIDB entry). For each set of GRIDB points with the same referenced fluid point the sort is further made on the angle (φ).
- d. The boundary point geometry table, generated above, is used at this time to form the boundary matrix part of the MATPØØL data block. For each entry in the table, all GRIDB points which reference it are appended to form a new entry of the following form.

## EXECUTIVE PREFACE MODULE IFP4 (INPUT FILE PROCESSOR - PHASE 4)

| <u>Field</u> | <u>Symbol</u>              | <u>Description</u>              |
|--------------|----------------------------|---------------------------------|
| 1            | $Id_f$                     | Fluid point identification      |
| 2-7          | $r, z, \ell, c, s, \rho_b$ | Fluid point properties          |
| 8            | $Id_g(1)$                  | GRIDB identification            |
| 9            | $\phi_1$                   | Angular position of GRIDB point |
| .            | .                          | .                               |
| .            | .                          | .                               |
| 6+2M         | $Id_g(M)$                  | GRIDB identification            |
| 7+2M         | $\phi_M$                   | Angular position of GRIDB point |
| 8+2M         | -1                         | End of entry flags              |
| 9+2M         | -1                         |                                 |

where M equals the number of GRIDB points that reference  $Id_f$ . Each new boundary fluid point entry is then placed in the MATP00L data block. The list of harmonic indices  $n_j$ , the gravity G, the N0SYM flag, the fluid coordinate system  $CS_f$ , and the symmetric boundary information (from the FLSYM card image) are placed in the first record of the boundary list data in the MATP00L data block. If DMIG data card images resulting from DMIAX data cards are present on the MATP00L data block, they are merged to the existing DMIG card image data. Matrix names are checked for uniqueness.

8. The RINGFL data cards define a ring (fluid point) with its axis coincidental with the axis of the fluid coordinate system. Its degrees of freedom are the harmonics of the pressure around the circle. Special GRID point card images corresponding to the RINGFL data cards are generated at this time and added to the GRID card images now on GE0M1. Each RINGFL card image is read and N GRID card images are created containing the following data.



# MODULE FUNCTIONAL DESCRIPTIONS

| <u>Field</u> | <u>Value</u>                      | <u>Description</u>                 |
|--------------|-----------------------------------|------------------------------------|
| 1            | $Id_f + 5 \cdot 10^5 \cdot (I_i)$ | Point identification               |
| 2            | $CS_f$                            | Fluid coordinate system number     |
| 3-5          | $X1, X2, X3$                      | Location coordinates               |
| 6            | -1                                | Fluid point flag                   |
| 7            | 0                                 | Permanent single point constraints |
| 8            | 0                                 | Not used                           |

where  $i$  goes from 1 to  $N$  for each RINGFL card image read.

SEQGP card images are created for each RINGFL card image and merged with SEQGP cards on GEØM1. The contents of the entry are:

| <u>Field</u> | <u>Value</u>                      | <u>Description</u>  |
|--------------|-----------------------------------|---------------------|
| 1            | $Id_f + 5 \cdot 10^5 \cdot (I_i)$ | Grid identification |
| 2            | $Id_f \cdot 10^3 + (I_i - 1)$     | Sequence number     |

where  $i$  goes from 1 to  $N$  for each RINGFL data card image.

## 4.89.7.2 GEØM2 Data Block Processing

1. The fluid element connections as specified by the CFLUID2, CFLUID3, and CFLUID4 card images are now operated upon. Each input card image is used together with the harmonic indices to define  $N$  "structural elements". The data given by the input card image is:

| <u>Field</u> | <u>Value</u> | <u>Description</u>                                      |
|--------------|--------------|---|
| 1            | $Id_e$       | Element identification number                           |
| 2 thru $j+1$ | $Id_j$       | Where $j = 2, 3, \text{ or } 4$ fluid point connections |
| $j+2$        | $\rho$       | Fluid density   |
| $j+3$        | $B$          | Fluid bulk modulus                                      |

For each input card image, connection card images are created for all harmonics in the problem. Their format is:



EXECUTIVE PREFACE MODULE IFP4 (INPUT FILE PROCESSOR - PHASE 4)

| <u>Field</u> | <u>Value</u>                | <u>Description</u>                          |
|--------------|-----------------------------|---|
| 1            | $Id_e \cdot 10^3 + I_i$     | Converted element identification            |
| 2 thru j+1   | $Id_j + 5 \cdot 10^5 (I_i)$ | Where j = 2, 3, or 4 connected fluid points |
| j+2          | $\rho$                      | Fluid density                               |
| j+3          | B                           | Fluid bulk modulus                          |
| j+4          | n                           | Harmonic number                             |

where  $i = 1, 2, \dots, N$  and  $n$  is an integer such that,

$$\frac{I_i - 3}{2} < n \leq \frac{I_i - 1}{2} \quad (13)$$

The harmonic element connection card images are merged into the GEOM2 data block as they are generated.

2. FSLIST card images each define a sequential list of fluid (RINGFL) points on a free surface. The FREEPT card images input to IFP4 each define a point on the free surface where a displacement may be output. The following operations are a result of FSLIST and FREEPT card images data.

For each fluid point ( $IDF_j$ ) defined in a FSLIST card image, a three word entry is placed in core containing  $IDF_j$ ,  $IDF_{j+1}$ , and  $\rho$ . The subscript  $j$  indicates the respective IDF in the FSLIST card image list of IDFs. if  $IDF_j$  or  $IDF_{j+1}$  is represented by "AXIS" in the FSLIST card image, a minus one (-1) is used. If  $IDF_j$  is the last point in the list,  $IDF_{j+1}$  is set to -2. If the fluid density ( $\rho$ ) is not present (an integer -1) in the FSLIST card image, the default fluid density ( $\rho_d$ ) from the AXIF image is used. If both  $\rho$  and  $\rho_d$  are missing a user fatal error results.

A set of structural mass elements are generated for each of the entries just added to core. Each set represents all harmonics in the problem. Connection card images called CMFREE elements are created such that each element consists of the following:

# MODULE FUNCTIONAL DESCRIPTIONS

| Field | Symbol           | Description  |
|-------|------------------|--|
| 1     | Id               | Element Id = $10^6 k + I_i$  |
| 2     | Idg <sub>1</sub> | $IDF_{k,1} + 5 \cdot 10^5 I_i$   |
| 3     | Idg <sub>2</sub> | $IDF_{k,2} + 5 \cdot 10^5 I_i$   |
| 4     | $\gamma_j$       | ( $\rho$ times G) the weight density, where G = gravity from AXIF image              |
| 5     | n                | Integer harmonic number such that;<br>$\frac{I_i - 3}{2} < n \leq \frac{I_i - 1}{2}$ |

where  $I_i$  represents the  $i^{th}$  entry in the harmonic index list, and k is the index of the entry in the FSLIST table of entries. If  $IDF_{k,1} = -1$ ,  $IDF_{k,1}$  is set to  $IDF_{k,2}$ . If  $IDF_{k,2} = -1$ , then  $IDF_{k,2}$  is set to  $IDF_{k,1}$ . Both can not be -1 initially. Thus for each entry, k = 1 thru K (the total number of entries), CMFREE images are created for all harmonic indices ( $I_i$ ), i = 1 thru N. These CMFREE element entries are merged into the GEOM2 data block as were the CFLUID2, CFLUID3, and CFLUID4 card images.

## 4.89.7.4 GEOM4 Data Block Processing

1. If FREEPT (free surface displacement point) card images are present, and gravity as specified in the AXIF card image is nonzero, a multipoint constraint (MPC) is generated at this time along with a scalar point (SPØINT) having the same identification number ( $Id_p$ ) as specified by each FREEPT card image. As each FREEPT card is read an SPØINT card image is placed in core and the following MPC card image is merged into the MPC data of GEOM4:

EXECUTIVE PREFACE MODULE IFP4 (INPUT FILE PROCESSOR - PHASE 4)

|                             | Field    | Symbol    | Value                       | Description                         |
|-----------------------------|----------|-----------|-----------------------------|-------------------------------------|
|                             | 1        | SID       | 102                         | Set identification number           |
|                             | 2        | GID       | $Id_p$                      | FREEPT identification number        |
|                             | 3        | Comp      | 0                           | Component                           |
|                             | 4        | $A_1$     | $- \rho G $                 | Density times gravity               |
| Repeats for<br>$i=1$ to $N$ | $2+3i$   | $GID_i$   | $Id_f + 5 \cdot 10^5 (I_i)$ | Fluid point harmonic identification |
|                             | $3+3i$   | Comp      | 0                           | Component                           |
|                             | $4+3i$   | $A_{i+1}$ | $C_i$                       | Harmonic coefficient                |
|                             | $\vdots$ | $\vdots$  | $\vdots$                    | $\vdots$                            |
|                             | $5+3N$   | Flag      | -1                          | End of mage flag                    |
|                             | $6+3N$   | Flag      | -1                          |                                     |
|                             | $7+3N$   | Flag      | -1                          |                                     |

$I_i$  is a harmonic index, and  $n$  equals an integer such that;

$$\frac{I_i - 3}{2} < n \leq \frac{I_i - 1}{2}, \quad (14)$$

$$\text{then: } \left. \begin{aligned} C_i &= \cosine \frac{n\pi\phi}{180} & \text{if } I_i \text{ is even,} \\ C_i &= \sin \frac{n\pi\phi}{180} & \text{if } I_i \text{ is odd,} \end{aligned} \right\} \quad (15)$$

where:  $\phi$  is the angle given in the FREEPT card image.

- Additional MPC card images are created if PRESPT card images are present. For each PRESPT card image read, an SPØINT is added to the in-core list of SPØINTs, and the following MPC card image is merged onto GEØM4.



# MODULE FUNCTIONAL DESCRIPTIONS

|                            | Field  | Symbol    | Value                    | Description                             |
|----------------------------|--------|-----------|--------------------------|---|
|                            | 1      | SID       | 102                      | Set identification                      |
|                            | 2      | GID       | $Id_p$                   | PRESPT identification                   |
|                            | 3      | Comp      | 0                        | Component                               |
|                            | 4      | $A_1$     | -1.0                     | Coefficient                             |
| Repeat for<br>$i=1$ to $N$ | $2+3i$ | $GID_i$   | $Id_f+5 \cdot 10^5(I_i)$ | Connected fluid harmonic identification |
|                            | $3+3i$ | Comp      | 0                        | Component                               |
|                            | $4+3i$ | $A_{i+1}$ | $C_i$                    | Harmonic coefficient                    |
|                            | .      | .         | .                        | .                                       |
|                            | .      | .         | .                        | .                                       |
|                            | $5+3N$ | Flag      | -1                       | End of image flag                       |
|                            | $6+3N$ | Flag      | -1                       |   |
|                            | $7+3N$ | Flag      | -1                       |   |

$I_i$  is a harmonic index, and  $n$  equals an integer such that,

$$\frac{I_i - 3}{2} < n \leq \frac{I_i - 1}{2} \quad (16)$$

$$\left. \begin{aligned} C_i &= \cosine \frac{n\pi\phi}{180} && \text{if } I_i \text{ is even.} \\ C_i &= \sin \frac{n\pi\phi}{180} && \text{if } I_i \text{ is odd.} \end{aligned} \right\} \quad (17)$$

$\phi$  is the angle given in the PRESPT card image.

3. If any SPØINTs were placed in core as a result of the presence of FREEPT or PRESPT card data, they are merged with the existing scalar point data on a scratch file.
4. At this time, if any harmonics are specified, an MPCADD card image is generated for each unique set identification present in the MPC and MPCADD card images on GEØM4. This MPCADD card image will then contain the internal set identification and include the user set identification. Thus as the MPC and MPCADD card images are read from GEØM4, a list of the set identifications present is created in core. An MPCADD card is then generated for

# EXECUTIVE PREFACE MODULE IFP4 (INPUT FILE PROCESSOR - PHASE 4)

each unique set identification (Id) present. Its format is:

| <u>Field</u> | <u>Value</u>               | <u>Description</u>               |
|--------------|----------------------------|----------------------------------|
| 1            | $2 \cdot 10^8 + \text{Id}$ | Internal set identification      |
| 2            | Id                         | User set identification          |
| 3*           | 102                        | Generated MPC set identification |
| 3 or 4*      | -1                         | End of image flag                |

If any MPCADD card images are present on GEØM4, as a result of the user's specifications, their set identification (Id) in field one is modified to an internal set identification ( $2 \cdot 10^8 + \text{Id}$ ), and if any MPC card images have been internally generated for set 102, the 102 set identification is added to the list of included set identifications therein.

If any MPC card images have been created for set 102, the following MPCADD card is generated in any event so as to assure that the 102 set be included in the solution. The set identification used here ( $2 \cdot 10^8$ ), will be referenced in later computations if the user has not referenced any MPC constraint set.

| <u>Field</u> | <u>Value</u>   | <u>Description</u>               |
|--------------|----------------|----------------------------------|
| 1            | $2 \cdot 10^8$ | Set identification               |
| 2            | 102            | Generated MPC set to be included |
| 3            | -1             | End of image flag                |

- Should the user specify gravity G to be zero (0) on the AXIF card, it is assumed that the effects of gravity on the free surface are to be removed. To accomplish this, a single point constraint (SPC set 102) set is created at this time by IFP4.

For each fluid point  $\text{Id}_f$  not equal to minus one (-1) or minus two (-2) in the free surface list, an SPC1 card image is merged onto GEØM4 containing the constraint information for all harmonics of this point. Its format is the following:

\*Set identification 102 is inserted only if any MPC card images have been generated for set 102. If  $\text{Id} = 102$  a user fatal error is indicated.

# MODULE FUNCTIONAL DESCRIPTIONS

| <u>Field</u> | <u>Value</u>              | <u>Description</u>          |
|--------------|---------------------------|-----------------------------|
| 1            | 102                       | Set identification          |
| 2            | 0                         | Component to be constrained |
| 2+1          | $Id_f + 5 \cdot 10^5 I_i$ | Point to be constrained     |
| .            | .                         | .                           |
| .            | .                         | .                           |
| 2+i          | $Id_f + 5 \cdot 10^5 I_i$ | Point to be constrained     |
| .            | .                         | .                           |
| .            | .                         | .                           |
| 2+N          | $Id_f + 5 \cdot 10^5 I_N$ | Point to be constrained     |
| 3+N          | -1                        | End of image flag           |

$I_i$  is the  $i^{th}$  entry in the list of harmonic indices.

6. Analogous operations to those described in paragraph (4) of this section are performed at this time for existing SPC, SPC1 and SPCADD card images. The data is merged onto a scratch file and when complete, the scratch file is merged onto the GEOM4 data block.

## 4.89.8 Subroutines

### 4.89.8.1 Subroutine Name: IFP4A

1. Entry Point: IFP4A
2. Purpose: To write the first line of the user fatal error messages.
3. Calling Sequence: CALL IFP4A(NUM)

NUM - Message number minus 4030.

### 4.89.8.2 Subroutine Name: IFP4B

1. Entry Point: IFP4B
2. Purpose : To copy data from IFP data files up to and including a given record onto a scratch file. On option the data on the scratch file may be copied onto the original data block.



EXECUTIVE PREFACE MODULE IFP4 (INPUT FILE PROCESSOR - PHASE 4)

3. Calling Sequence:

CALL IFP4B(FILE, SCRT, ANY, SPACE, LSPACE, RECID, EOF)

where:

FILE - File Number  
SCRT - Scratch File Number  
ANY - Flag = .TRUE. if RECID is found on FILE,  
= .FALSE. if record is missing  
SPACE - Area of core for working space  
LSPACE - Length of working space  
RECID - Record Number of FILE where the copy process stops. If -1 is  
used the copy process proceeds to the end of FILE and the data  
on SCRT is copied back onto FILE.  
EOF - Flag = .TRUE. if end of record is encountered on return.

4.89.8.3 Subroutine Name: IFP4C

1. Entry Point: IFP4C

2. Purpose: To open an IFP generated file and a scratch file and to copy the header record  
from the IFP file onto the scratch file.

3. Calling Sequence:

CALL IFP4C(FILE, SCRT, BUF1, BUF2, EOF)

where:

FILE - File number  
SCRT - Scratch file number  
BUF1 - Buffer area in core for reading FILE  
BUF2 - Buffer area in core for writing SCRT  
EOF - Flag = .TRUE. if FILE is null

4.89.8.4 Subroutine Name: IFP4E

1. Entry Point: IFP4E

2. Purpose: To check identification numbers of fluid points for possible difficulties with  
large numbers.

## MODULE FUNCTIONAL DESCRIPTIONS

3. Calling Sequence: CALL IFP4E(ID)

where:

ID - Identification number

### 4.89.8.5 Subroutine Name: IFP4F

1. Entry Point: IFP4F
2. Purpose: To test if a bit in a trailer record word is on or off.
3. Calling Sequence: CALL IFP4F(IBIT,FILE,BIT)

where:

IBIT - Position of bit in trailer

FILE - File number

BIT - Flag = .TRUE. if bit is on  
= .FALSE. if bit is off

### 4.89.8.6 Subroutine Name: IFP4G

1. Entry Point: IFP4G
2. Purpose: To turn on a bit in a trailer record.
3. Calling Sequence: CALL IFP4G(IBIT,FILE)

where:

IBIT - Position of bit in trailer

FILE - File number

## FUNCTIONAL MODULE BMG (BOUNDARY MATRIX GENERATOR FOR HYDROELASTIC PROBLEMS)

### 4.90 FUNCTIONAL MODULE BMG (BOUNDARY MATRIX GENERATOR FOR HYDROELASTIC PROBLEMS)

#### 4.90.1 Entry Point: BMG

#### 4.90.2 Purpose

The MATP00L data block may contain data related to fluid boundaries which is generated by the IFP4 preface module. The purpose of this module is to combine these data with the geometry data (EQEXIN, BGPDT, and CSTM data blocks) to produce matrix terms which describe fluid-structure connection forces. These matrix terms are produced in the form of internal DMIG data card images. The module MTRXIN must always be used in conjunction with module BMG to produce NASTRAN matrices.

#### 4.90.3 DMAP Calling Sequence

BMG MATP00L,BGPDT,EQEXIN,CSTM / BDP00L / V,N,N0KBFL / V,N,N0ABFL / V,N,MFACT \$

#### 4.90.4 Input Data Blocks

MATP00L - Direct Input Matrices and Hydroelastic Boundary data.

BGPDT - Basic Grid Point Definition Table.

EQEXIN - Equivalence between External and Internal Grid Point numbers.

CSTM - Coordinate System Transformation Matrices.

#### 4.90.5 Output Data Blocks

BDP00L - Boundary Matrices ABFL and KBFL in DMIG Format.

#### 4.90.6 Parameters

N0KBFL - Existence of KBFL Matrix Data = 0,  
No KBFL Data = -1, output parameter.

N0ABFL - Existence of ABFL Matrix Data = 0,  
No ABFL Data = -1, output parameter.

MFACT - Complex Factor for Symmetric Structures, output parameter.

#### 4.90.7 Method

The fluid boundary data, contained in the MATP00L data block, is grouped by the fluid points on the boundary. For each fluid point the geometric parameters of the surface and the positions of the associated grid points are listed. The input data read for each fluid point are operated



# FUNCTIONAL MODULE BMG (BOUNDARY MATRIX GENERATOR FOR HYDROELASTIC PROBLEMS)

on to produce matrix terms. The output matrix data are written on two files. The ABFL matrix terms are written on the BDPØØL file. The KBFL data are written on a scratch file. After the processing of the input file is complete, the KBFL data is appended to the BDPØØL file.

During the processing the core is allocated for the geometry data blocks. For each fluid point, tables are also created which must fit in the remaining core. The following description lists the form of the input data and the various steps used in the process.

## 4.90.7.1 Form of the Boundary Data Record on the MATPØØL Data Block

Three levels of data are contained within this record. The first level is a list of the overall definition parameters of the fluid and the boundary. The second level consists of fluid points and their associated data. Attached to each fluid point is a third level consisting of a list of the connected structural grid points and their angular position on the fluid circle. The actual data in the record are:

### 1. Header data:

| <u>Symbol</u>                      | <u>Description</u>                                |
|------------------------------------|---|
| $CS_f$                             | Fluid coordinate system identification            |
| M                                  | Number of symmetric sections                      |
| S1<br>S2                           | Symmetry definitions of first and second boundary |
| g                                  | Value of gravity                                  |
| NØSYM                              | Flag for n* series                                |
| k                                  | Number of harmonic indices below                  |
| $n_1, n_2, \dots, n_j, \dots, n_k$ | List of harmonic indices                          |

# MODULE FUNCTIONAL DESCRIPTIONS

## 2. Fluid point data

| <u>Symbol</u>            | <u>Description</u>                        |
|--------------------------|---|
| $Id_{f1}$                | First fluid point (RINGFL) identification |
| $r, z, \ell, C, S, \rho$ | Fluid point properties                    |
| $N_g$                    | Number of connected grid points below     |
| $Id_1$                   | Grid point identification                 |
| $\phi_1$                 | Angular position of First grid point      |
| $Id_2$                   | ⋮   |
| $\phi_2$                 | ⋮   |
| ⋮                        | ⋮   |
| $Id_i$                   | ⋮   |
| $\phi_i$                 | (grid point data)                         |
| ⋮                        | ⋮   |
| $Id_g$                   | ⋮   |
| $\phi_g$                 | ⋮   |
| $Id_{f2}$                | Second fluid point                        |
| ⋮                        | ⋮   |
| etc.                     | etc.                                      |

## 4.90.7.2 Selection of Harmonics to Match Boundary Conditions

The Header Record for the boundary data is read and a list of harmonics ( $n_j$  and  $n_j^*$ ) to be included in the matrices are precalculated. If  $NOSYM = YES$ , the indices for the sine series will be included. A test is made on each value of  $n$  and  $n^*$  using the values of  $S1$  and  $S2$  in the header data.

1. If  $M = 0$  or 1 accept all values of  $n$  and  $n^*$ .

2. If  $S1 = S2$ , Calculate:

$$K = \frac{2n}{M} \quad (1)$$

a. If  $K$  is not an integer reject  $n$  or  $n^*$

b. If  $K$  is an integer:

accept  $n$  if  $S1 = S$

accept  $n^*$  if  $S1 = A$

FUNCTIONAL MODULE BMG (BOUNDARY MATRIX GENERATOR FOR HYDROELASTIC PROBLEMS)

3. If  $S1 \neq S2$ , calculate:

$$K = \frac{1}{2} \left[ \frac{4n}{M} - 1 \right] \quad (2)$$

a. If  $K$  is not an integer reject  $n$  or  $n^*$

b. If  $K$  is an integer:

accept  $n$  if  $S1 = S$

accept  $n^*$  if  $S1 = A$

A list of allowable values of  $n$  and  $n^*$  is built in core. If the final list is null, only the KBFL matrix is generated. The parameter MFACT is the complex number  $(M,0)$  if  $M$  is nonzero. The value  $(1,0)$  is used if  $M$  is zero.

#### 4.90.7.3 Formation of Geometry Table for Internal Use

The core is allocated for the BGPDT data block and an extra word for each of its entries. The data is read in groups of four words and stored in five word entries, reserving the first word for the external identification number. The EQEXIN data contains a paired list of external and internal numbers. The EQEXIN is read and the external numbers are placed in the corresponding BGPDT entries in core. The resulting BGPDT data are sorted on the external identification numbers. The referenced coordinate system number and the basic location vector for any grid point are now available by using a single binary search.

#### 4.90.7.4 CSTM Processing

The CSTM data block is now read and stored in core. The fluid coordinate system identification number,  $CS_f$ , is found and the 3 by 3 transformation matrix,  $[T_{of}]$ , is extracted directly from the data.

#### 4.90.7.5 File Initialization

The processing of the matrix data may now begin. The files for the ABFL and KBFL output data are opened and the matrix header data is written. The boundary data on the MATPØØL data block are read and one fluid point at a time is processed.



# MODULE FUNCTIONAL DESCRIPTIONS

## 4.90.7.6 Calculations of Areas Associated with Boundary Grid Points

The first set of the parameters  $Id$ ,  $r$ ,  $z$ ,  $\rho_k$ ,  $c_k$ ,  $S_k$ , and  $\rho_k$  are read for the fluid point where  $k = 1$  if the fluid point has only one entry or  $k = 1, 2, 3 \dots$  if the fluid point is connected with multiple boundaries. The connected grid point numbers ( $Id_i$ ) and angles ( $\phi_i$ ) are read and placed in core. Twenty-six words are allocated for each grid point.

For each connected grid point the calculated data are:

|         |             |                                     |
|---------|-------------|-------------------------------------|
| (1)     | $Id_i$      | Identification number               |
| (2)     | $\phi_i$    | Azimuthal angle (radians)           |
| (3)     | $\phi_{0i}$ | Angle midway to previous point      |
| (4)     | $\phi_{1i}$ | Angle midway to next point          |
| (5-22)  | $[V_i]$     | 3x3 double precision transformation |
| (23-26) | $\{W_i\}$   | 3x1 vector                          |

The midway angles are defined in general as:

$$\begin{aligned}\phi_{0i} &= \frac{\phi_i + \phi_{i-1}}{2} , \\ \phi_{1i} &= \frac{\phi_i + \phi_{i+1}}{2} .\end{aligned}\tag{3}$$

The angles for the first point are:

$$\begin{aligned}\phi_{01} &= \phi_1 , \quad M \geq 2 , \\ \phi_{01} &= \frac{\phi_1 + \phi_N - 2\pi}{2} , \quad M = 0.\end{aligned}\tag{4}$$

The angles for the last point are:

$$\begin{aligned}\phi_{1N} &= \phi_N , \quad M \geq 2 , \\ \phi_{1N} &= \frac{\phi_N + \phi_1 + 2\pi}{2} , \quad M = 0 .\end{aligned}\tag{5}$$

All of the grid point data are sorted on the grid point numbers before the transformations  $[V_i]$  and  $\{W_i\}$  are calculated.

# FUNCTIONAL MODULE BMG (BOUNDARY MATRIX GENERATOR FOR HYDROELASTIC PROBLEMS)

The equations for the transformation matrices are:

$$\begin{aligned} [V_i] &= [T_i]^T [T_{of}] , \\ \{W_i\} &= [T_i]^T [T_{of}] \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix} , \end{aligned} \quad (6)$$

Where  $[T_i]$  is the 3 by 3 global-to-basic transformation matrix for grid point  $i$ .

## 4.90.7.7 Calculation of Matrix Terms

The matrix terms corresponding to one fluid point are generated for the ABFL and KBFL data tables at this stage. The ABFL matrix terms are generated as follows:

1. For each allowable value of  $n$  or  $n^*$  a matrix column is generated in the ABFL table. The internal numbers,  $G_j$ , for the fluid point identification number,  $Id_f$ , are:

$$G_j = Id_f + (n + 1)10^6 \quad \text{cosine series}$$

$$G_j = Id_f + \frac{2n^* + 1}{2} 10^6 \quad \text{sine series}$$

These numbers label the column of the matrix.

2. Each row position in the matrix is labeled by a grid point,  $Id_i$ , and its three components:  $C = 1, 2, 3$ . The values corresponding to these positions are the values of the vector  $\{A_{in}\}$  where:

$$\{A_{in}\} = \sum_k A_{ik}^n [V_i] \begin{Bmatrix} C_k \cos \phi_i \\ C_k \sin \phi_i \\ S \end{Bmatrix} . \quad (7)$$

The vector  $\{A_{in}^*\}$  is similar. The coefficients are:

$$\begin{aligned} A_{ik}^0 &= r_{\ell k} (\phi_{1i} - \phi_{0i}) , \quad n = 0 \\ A_{ik}^n &= \frac{r_{\ell k}}{n} [\sin(n\phi_{1i}) - \sin(n\phi_{0i})] , \quad n > 0 \\ A_{ik}^{n^*} &= \frac{r_{\ell k}}{n} [\cos(n^*\phi_{1i}) - \cos(n^*\phi_{0i})] , \quad n^* > 0 \end{aligned} \quad (8)$$

## MODULE FUNCTIONAL DESCRIPTIONS

where  $k = 1, 2, 3, \dots$  is an index if the fluid point occurs more than once in the boundary list tables.

3. If gravity,  $g$ , is nonzero the KBFL matrix terms are calculated. Each grid point,  $Id_i$ , connected to the fluid point is used to produce three columns of the matrix corresponding to the three components,  $C_j = 1, 2, 3$ . The three rows for each column are the same three components,  $C_i = 1, 2, 3$ , of the grid point. The equation for the three terms in each column is:

$$\{K_i\}_j = \sum_k B_{ii}^k W_{ij} [V_i] \begin{Bmatrix} C_k \cos \phi_i \\ C_k \sin \phi_i \\ S_k \end{Bmatrix} \quad (9)$$

where:

$W_{ij}$  is the  $j^{\text{th}}$  component of  $\{W_i\}$  and:

$$B_{ii}^k = r_{ik}^2 \rho_k (\phi_{i1} - \phi_{i0}) g. \quad (10)$$

$k$  is the index used if the points are included in more than one boundary list entry.

### 4.90.7.8 Wrapup Operation

The final operations involve rewinding the scratch file containing the KBFL data and appending that data to the BDP00L data block output file.

### 4.90.8 Additional Subroutines

BMGTNS - This routine is a slightly modified version of utility subroutine PRETRD so as to have only one entry point.

### 4.90.9 Design Requirements

The major core requirements are that the BGPDT data must fit and that twenty-six words for each boundary grid point converted to one fluid point must fit in core.



FUNCTIONAL MODULE BMG (BOUNDARY MATRIX GENERATOR FOR HYDROELASTIC PROBLEMS)

4.90.10 Diagnostic Messages

The following system fatal messages may be issued by BMG:

\*\*\*SYSTEM FATAL MESSAGE 2148, CØØRDINATE SYSTEM = XXXXX CAN NØT BE FOUND IN CSTM DATA.

\*\*\*SYSTEM FATAL MESSAGE 2149, CØNNECTED FLUID PØINT ID = XXXXX IS MISSING BGPDT DATA.

## EXECUTIVE PREFACE MODULE IFP5 (INPUT FILE PROCESSOR - PHASE 5)

### 4.91 EXECUTIVE PREFACE MODULE IFP5 (INPUT FILE PROCESSOR - PHASE 5)

#### 4.91.1 Entry Point: IFP5

#### 4.91.2 Purpose

1. To convert the data card images related to acoustic analysis into conventional grid points and elements.
2. To calculate slot-cavity interface matrix terms and generate corresponding scalar elements.
3. To generate plot elements describing the sides of the acoustic elements.

#### 4.91.3 Calling Sequence

CALL IFP5. IFP5, an Executive Preface Module, is called only by the Preface driver SEMINT.

#### 4.91.4 Input Data Blocks

AXIC - Contains Bulk Data Cards related to the acoustic parameter, points, and boundaries.

GEØM1- Grid point and coordinate system data

GEØM2- Element Data, including acoustic elements.

#### 4.91.5 Output Data Blocks

GEØM1 - Same format as input, acoustic points are merged in as GRID points.

GEØM2 - Same format as input, scalar elements and plot elements are added.

#### 4.91.6 Parameters

Not applicable to IFP5

#### 4.91.7 Method

IFP5 converts the data on the AXIC data block into conventional grid points and elements. The GEØM1 and GEØM2 data are read and merged with the new data on scratch files. The complete data sets are copied back onto the GEØM1 and GEØM2 files. (This is not normally allowed in a NASTRAN Module. The preface modules, however, have the privilege of writing on an input file.)

The data cards listed below are processed by IFP5. The corresponding output card image and its data block are given for each card.

## EXECUTIVE PREFACE MODULE IFP5 (INPUT FILE PROCESSOR - PHASE 5)

| IFP5 Input<br>Card Image                                 | Data Block<br>In | IFP5 Output<br>Card Image  | Data Block<br>Out |
|--|------------------|--|-------------------|
| AXSLØT   | AXIC             | (all below)  | (all below)       |
| CAXIF2 }<br>CAXIF3 }<br>CAXIF4 }<br>CSLØT3 }<br>CSLØT4 } | GEØM2            | { CAXIF2 }<br>{ CAXIF3 }<br>{ CAXIF4 }<br>{ CSLØT3 }<br>{ CSLØT4 }<br>{ PLØTEL } | GEØM2             |
| GRIDF }<br>GRIDS }                                       | AXIC             | GRID   | GEØM1             |
| BDYLIST  | AXIC             | CELAS2   | GEØM2             |

It should be noted that the formats of the CAXIFi data cards are exactly the same as the CFLUIDi data cards as generated by IFP4, Section 4.89. The following steps are followed to process the data:

1. The AXSLØT card is read from the AXIC file, its data are:

$\rho_d$  - default density  
 $B_d$  - default bulk modulus  
 $N$  - harmonic number  
 $w_d$  - default slot width  
 $M$  - Number of slots

2. The GRIDS data card images are read and stored in core. The contents of each card are:

$Id_s$  - identification number  
 $r$  - radius  
 $z$  - axial coordinate  
 $w$  - slot width  
 $Id_f$  - identification of assoicated GRIDF

3. The GRIDF data card images are read and stored in core. The contents of each card are

$Id_f$  - identification number  
 $r$  - radius  
 $z$  - axial coordinate



# MODULE FUNCTIONAL DESCRIPTIONS

4. If the value in field 5 of GRIDS card is nonzero an IDF card is generated with the values  $Id_f$ ,  $r$ ,  $z$  as given on the GRIDS card. These cards are added to the GRIDF images and the complete list of GRIDF cards is sorted.
5. Data block GEØM1 is copied onto the first scratch file up to the first GRID card. The GRIDF and GRIDS cards are merged with the GRID cards in the GRID card format as follows:

| <u>GRID Field</u> | <u>Value GRIDF</u> | <u>Value GRIDS</u> |
|-------------------|--------------------|--------------------|
| 1                 | $Id_f$             | $Id_s$             |
| 2                 | 0                  | 0                  |
| 3                 | $r$                | $r$                |
| 4                 | $z$                | $z$                |
| 5                 | 0                  | $w$                |
| 6                 | -1                 | -1                 |
| 7                 | 0                  | 0                  |

6. The remainder of GEØM1 is copied onto the scratch file. The scratch file is then copied back onto GEØM1, starting from the beginning.
7. The SLBDY data card images are read from the AXIC data block. For each entry,  $Id_i$  on a logical card, five words are allocated in core and the following is stored.

$$Id_i, Id_{i-1}, Id_{i+1}, RHØ, M$$

where  $Id_i$  is a point number in the list

$Id_{i-1}$  is the previous point number in the list

$Id_{i+1}$  is the next point number in the list

$RHØ, M$  are given on the logical card

If  $Id_i$  is the first entry on a logical card,  $Id_{i-1} = -1$ . If  $Id_i$  is the last entry on a logical card,  $Id_{i+1} = -1$ .

8. After all SLBDY cards are processed the above list is sorted on the first entry in each group of 5.
9. Plot elements (PLØTEL) are generated and placed on the first scratch file. The GEØM2 data block is read and for each (AXIFI) data card a series of PLØTEL cards are generated and written on the first scratch file (SCRT1).

## EXECUTIVE PREFACE MODULE IFP5 (INPUT FILE PROCESSOR - PHASE 5)

CAXIF2 Data

Id  
G1  
G2  
 $\rho$   
B  
N

PLØTEL Data

Id +  $10^6$   
G1  
G2

CAXIF3 Data

Id  
G1  
G2  
G3  
 $\rho$   
B  
N

PLØTEL Data

Id +  $2 \cdot 10^6$   
G1  
G2  
  
Id +  $3 \cdot 10^6$   
G2  
G3  
  
Id +  $4 \cdot 10^6$   
G3  
G1

CAXIF4 Data

Id  
G1  
G2  
G3  
G4  
 $\rho$   
B  
N

PLØTEL Data

|                         |                         |
|-------------------------|-------------------------|
| Id + $5 \cdot 10^6$     | Id + $6 \cdot 10^6$     |
| G1                      | G2                      |
| G2                      | G3                      |
| <br>Id + $7 \cdot 10^6$ | <br>Id + $8 \cdot 10^6$ |
| G3                      | G4                      |
| G4                      | G1                      |

10. A second scratch file (SCRT2) is opened and the GEØM2 data is copied onto SCRT2 to the CELAS2 data card position. The boundary list data is processed and CELAS2 data cards are generated and appended to SCRT2. For each five word entry in the Boundary Table search the GRIDS data card images for the following data

$r_i, z_i, w_i, IDF$  from GRIDS card "IDS<sub>i</sub>"

$r_{i-1}, z_{i-1}, w_{i-1}$  from GRIDS card "IDS<sub>i-1</sub>"

$r_{i+1}, z_{i+1}, w_{i+1}$  from GRIDS card "IDS<sub>i+1</sub>"

where  $r = r_i, z = z_i$  if the corresponding identification number IDS is -1. If a GRIDS card can not be found a fatal error exists. The following data is calculated for each entry:

# MODULE FUNCTIONAL DESCRIPTIONS

$$\ell_1 = \sqrt{(z_{i+1} - z_i)^2 + (r_{i+1} - r_i)^2} ,$$

$$\ell_2 = \sqrt{(z_{i-1} - z_i)^2 + (r_{i-1} - r_i)^2} ,$$

$$\bar{\ell} = \frac{1}{2} \sqrt{(z_{i+1} - z_{i-1})^2 + (r_{i+1} - r_{i-1})^2} ,$$

$$\bar{w} = \frac{1}{4(\ell_1 + \ell_2)} [\ell_1 w_{i+1} + \ell_2 w_{i-1}] + \frac{3}{4} w_i ,$$

$$\bar{r} = \frac{1}{4(\ell_1 + \ell_2)} [\ell_1 r_{i+1} + \ell_2 r_{i-1}] + \frac{3}{4} r_i .$$

The coefficients for slot interaction are:

$$\beta = \frac{2\pi\bar{r}}{M\bar{w}}$$

(If  $\beta \geq 1$  a fatal error exists)

$$\ell_c = \frac{\bar{w}}{2\pi} \left[ \left( \beta + \frac{1}{\beta} \right) \log_e \left( \frac{\beta+1}{\beta-1} \right) + 2 \log_e \frac{(\beta+1)(\beta-1)}{\beta r} \right]$$

and

$$\ell_e = \text{Max} (\ell_c, .01\bar{w})$$

$$K_f = \frac{\bar{w} \bar{\ell}}{\rho \ell_e} F_i$$

where  $F_i = M$  if  $N = 0$  or  $N = \frac{M}{2}$ ,  $G_i = \frac{M}{2}$  otherwise.

11. For each entry in the Boundary Table, corresponding GRIDF data card with ID = IDF is found in core. For the corresponding GRIDF point  $IDF_j$  calculate the following:

$$\alpha = \frac{\sin \frac{N\bar{w}}{2\bar{r}}}{\frac{N\bar{w}}{2\bar{r}}}$$



EXECUTIVE PREFACE MODULE IFP5 (INPUT FILE PROCESSOR - PHASE 5)

12. CELAS2 elements are now generated for the slot point  $IDS_i$  and the corresponding axisymmetric fluid point  $IDF_j$ . The format of this data is:

| Id         | K                     | G1      | C1  | G2      | C2    |
|------------|-----------------------|---------|-----|---------|-------|
| $Id_e + 1$ | $(1-\alpha)K_f$       | $IDS_i$ | "1" | blank   | blank |
| $Id_e + 2$ | $\alpha K_f$          | $IDS_i$ | "1" | $IDF_i$ | "1"   |
| $Id_e + 3$ | $\alpha(1-\alpha)K_f$ | $IDF_i$ | "1" | blank   | blank |

The element identification numbers  $Id_e$  are sequential starting with 10,000,001. With each new point on the boundary list,  $IDF_i$ , the value  $Id_e$  is incremented by 3.

13. When all points on the boundary list are processed, the remainder of GEØM2 is copied onto SCRT2. If CSLØT3 and/or CSLØT4 elements are encountered, they will produce PLØTEL data card images which are written on SCRT1 in the following format:

CSLØT3 Data

Id  
G1  
G2  
G3  
 $\rho$   
B  
M  
N

PLØTEL Data

$Id + 9 \cdot 10^6$   
G1  
G2  
 $Id + 10 \cdot 10^6$   
G2  
G3  
 $Id + 11 \cdot 10^6$   
G3  
G1

CSLØT4 Data

Id  
G1  
G2  
G3  
G4  
 $\rho$   
B  
M  
N

PLØTEL Data

$Id + 12 \cdot 10^6$   
G1  
G2  
 $Id + 13 \cdot 10^6$   
G2  
G3  
 $Id + 14 \cdot 10^6$   
G3  
G4  
 $Id + 15 \cdot 10^6$   
G4  
G1

## MODULE FUNCTIONAL DESCRIPTIONS

14. When GEØM2 has been completely copied onto SCRT2, the files are rewound and the data from SCRT2 (containing the new CELAS2 elements) and the PLØTEL data from SCRT1 are merged and copied back onto GEØM2.

### 4.91.8 Subroutines

#### 4.91.8.1 IFP5A

1. Entry Point: IFP5A
2. Purpose: Prints formal part of messages for IFP5.
3. Calling Sequence: CALL IFP5A (NUM)  
NUM = IFP5 message number.

### 4.91.9 Design Requirements

Discussed under Method.

### 4.91.10 Diagnostic Messages

Many user messages relevant to the acoustic cavity modeling data may be issued.

## FUNCTIONAL MODULE PLTTRAN

### 4.92 FUNCTIONAL MODULE PLTTRAN

#### 4.92.1 Entry Point: PLTTRA

#### 4.92.2 Purpose

To modify the SIL and BGPDT tables for the purpose of plotting special scalar grid points. Each grid point with one degree of freedom is given six degrees of freedom in the modified SIL data block. These points are identified in the BGPDP data block by the value (-2) in the first entry for each point.

#### 4.92.3 DMAP Calling Sequence

PLTTRAN BGPDT,SIL / BGPDP,SIP / V,N,LUSET / V,N,LUSEP \$

#### 4.92.4 Input Data Blocks

Data Block BGPDT - Four entries per grid or scalar point as follows:

1. Local coordinate system number or -1 if point is a scalar point.
- 2-4. X, Y, Z location in basic coordinate system.

Data Block SIL - One entry per grid or scalar point. The value of the entry is the location of the first degree of freedom of the point in the vector containing all degrees of freedom.

#### 4.92.5 Output Data Blocks

Data Block BGPDP - Same format as BGPDT. If a point is determined to have one degree of freedom and is not a scalar point, the value (-2) is placed in the first entry for that point.

Data Block SIP - Same format as SIL. All points except time scalar points are given six (6) degrees of freedom. A true scalar point has the value (-1) in the first slot of its BGPDT entry.

#### 4.92.6 Parameters

LUSET - Total number of degrees of freedom

LUSEP - New value for LUSET taking into account the change in the number of degrees of freedom when the special scalar points are expanded to six degrees of freedom.



## MODULE FUNCTIONAL DESCRIPTIONS

### 4.92.7 Method

The SIL is read 1 word at a time; the BGPDT is read 4 words at a time. If the difference between the new SIL number and the previous SIL value is 1 and the first entry in the BGPDT is zero a fluid scalar grid point exists. In this event the new SIP increment is 6 and the value -2 is placed in the first word of the BGPDP entry.

### 4.92.8 Subroutines

None.

### 4.92.9 Design Requirements

Open core is defined at /PLTRN1/ and must be sufficient to hold four (4) GINØ buffers.

### 4.92.10 Diagnostic Messages

Messages 3001, 3002, 3003, 3008, 5011 and 5012 may be issued.

## MATRIX MODULE UPARTN

### 4.93 MATRIX MODULE UPARTN (PARTITIONS A MATRIX BASED ON USET)

#### 4.93.1 Entry Point: DUPART

#### 4.93.2 Purpose:

To compute a partitioning vector based on the displacement sets as defined by USET and create the symmetric partitions of the input matrix.

For example this module will perform

$$[K_{nn}] \Rightarrow \begin{bmatrix} K_{ff} & K_{fs} \\ K_{sf} & K_{ss} \end{bmatrix}$$

#### 4.93.3 DMAP Calling Sequence

UPARTN USET,KNN / KFF,KSF,KFS,KSS / C,Y,MAJOR=N / C,Y,SUBO=F / C,Y,SUBI=S \$

#### 4.93.4 Input Data Blocks

USET - Displacement set definitions table (This may also be USETD if extra points are present).

KNN - Any square displacement matrix. The associated set of KNN (N) must be given in the first parameter.

Note: 1. USET may not be purged.

2. If KNN is purged, UPARTN will return.

#### 4.93.5 Output Data Blocks

KFF - Matrix. It will have SUBO rows and columns.

KSF - Matrix. It will have SUBI rows and SUBO columns.

KFS - Matrix. It will have SUBO rows and SUBI columns.

KSS - Matrix. It will have SUBI rows and columns.

Note: Any purged or omitted output data blocks will not be written.

## MODULE FUNCTIONAL DESCRIPTIONS

### 4.93.6 Parameters

MAJØR - Input - BCD - No default value. This is the set of KNN.

SUB0 - Input - BCD - No default value. This is the first subset of MAJØR.

SUB1 - Input - BCD - No default value. This is the second subset of MAJØR.

Note: 1. MAJOR, SUB0, and SUB1 must be selected from the following list: M,Ø,R,SG,SB,L,  
A,F,S,N,G,E,P,NE,FE, and D.

2. The set equation  $MAJØR = SUB0 + SUB1$  should be satisfied.

### 4.93.7 Method

The module driver, DUPART, checks the compatibility of the parameter data and directly calls UPART and MPART (an entry point in UPART). All work is then accomplished in the UPART routine.

### 4.93.8 Subroutines

UPART - See subroutine description, section 3.5.9.

### 4.93.9 Design Requirements

One scratch file.

### 4.93.10 Diagnostic Messages

UPARTN may issue one of the following diagnostic messages:

3007 or 3059



## MATRIX MODULE UMERGE

4.94 MATRIX MODULE UMERGE (MERGES TWO MATRICES BASED ON USET).

4.94.1 Entry Point: DUMERG

4.94.2 Purpose:

To merge two matrices into a third based on the displacement sets. For example, this module will perform:

$$\begin{Bmatrix} \phi_a \\ \phi_o \end{Bmatrix} \Rightarrow \{ \phi_f \}$$

4.94.3 DMAP Calling Sequence

UMERGE USET,PHIA,PHIØ / PHIF / C,Y,MAJØR=F / C,Y,SUBO=A / C,Y,SUB1=Ø \$

4.94.4 Input Data Blocks

USET - Displacement set definitions table (this may also be USETD if extra points are present.)

Any two matrices except that their rows must be associated with degrees-of-freedom

PHIA  
PHIØ - specified by USET and the parameter list. PHIA's degrees-of-freedom are specified by SUBO and PHIØ's by SUB1.

Note: Either matrix may not be present and its respective degrees-of-freedom will be filled with zeros.

4.94.5 Output Data Blocks

PHIF - Matrix. Its terms will be associated with degrees-of-freedom in the set specified by MAJØR.

Note: PHIF must be present.

4.94.6 Parameters

MAJØR - Input - BCD - No default value. This is the set of PHIF.

SUBO - Input - BCD - No default value. This is the set of PHIA.

SUB1 - Input - BCD - No default value. This is the set of PHIØ.

## MODULE FUNCTIONAL DESCRIPTIONS

- Note: 1. MAJOR, SUB0, and SUB1 must be selected from the following list: M,Ø,R,SG,SB,L,A,  
F,S,N,G,E,P,NE,FE and D.
2. The set equation  $MAJØR = SUB0 + SUB1$  should be satisfied.

### 4.94.7 Method

The module driver, DUMERG, checks the compatibility of the parameter data and directly calls SDR1B. All work is then accomplished in the SDR1B routine.

### 4.94.8 Subroutines

SDR1B - See its subroutine description, section 3.5.

### 4.94.9 Design Requirements

One scratch file.

### 4.94.10 Diagnostic Messages

UMERGE may issue one of the following diagnostic messages:

3007 or 3059

## MATRIX MODULE VEC

### 4.95 MATRIX MODULE VEC (CREATES PARTITIONING VECTOR BASED ON USET)

#### 4.95.1 Entry Point: VEC

#### 4.95.2 Purpose:

VEC creates a partitioning vector based on USET that may be used in PARTN and MERGE.

#### 4.95.3 DMAP Calling Sequence

VEC     USET / V / C,N,SET / C,N,SET0 / C,N,SET1 \$

#### 4.95.4 Input Data Blocks

USET - Displacement set definition table (this may be USETD if extra points are present).

Note: USET must be present.

#### 4.95.5 Output Data Blocks

V - Partitioning vector.

Note: V may not be purged.

#### 4.95.6 Parameters

SET - Input-BCD-no default. SET indicates the set to which the partitioning vector applies.

SET0 - Input-BCD-no default. SET0 indicates the upper partition of SET.

SET1 - Input-BCD-no default. SET1 indicates the lower partition of SET.

#### 4.95.7 Method

The BCD parameters SET, SET0, and SET1 are converted to bit positions in USET. They must be one of the following 17 symbols: M,Ø,R,SG,SB,L,A,F,S,N,G,E,P,NE,FE,D,H or else a fatal error will result.

USET is read into core and the file closed. The output file is then opened and each entry is compared with the three converted parameters as follows:

1. USET is searched for members of SET. If the entry is not a member of SET, it is checked that it is not a member of SET0 or SET1 before going to the next entry.



## MODULE FUNCTIONAL DESCRIPTIONS

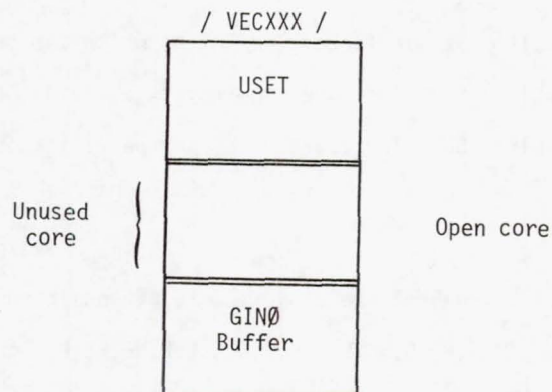
2. The entry that belongs to SET is then checked if it is also a member of SET1. If it is, the entry is also checked if it is a member of SET0, which is fatal, before replacing the entry with 1.0.
3. If the entry is a member of SET and not a member of SET1, the entry is checked to verify that it is a member of SET0 before replacing it with a 0.0.
4. After all entries have been successfully processed, a check is made to insure that a vector exists and that the entries are not all zeros or ones (fatal error).
5. The rewritten entries are then written onto the output data block as a matrix consisting of one (1) column.

### 4.95.8 Subroutines

VEC has no auxiliary subroutines.

### 4.95.9 Design Requirements

1. Open core is defined at / VECXXX /
2. Layout of open core is as follows:



## MATRIX MODULE ADD5 (ADD MATRICES)

### 4.96 MATRIX MODULE ADD5 (ADD MATRICES)

#### 4.96.1 Entry Point: DADD5

#### 4.96.2 Purpose

To compute  $[X] = \alpha[A] + \beta[B] + \gamma[C] + \delta[D] + \epsilon[E]$ .

#### 4.96.3 DMAP Calling Sequence

```
ADD  A,B,C,D,E/X/C,Y,ALPHA=(1.0,2.0)/C,Y,BETA=(3.0,4.0)/C,Y,GAMMA=(5.0,6.0)/  
      C,Y,DELTA=(7.0,8.0)/C,Y,EPSLN=(9.0,0.0) $
```

#### 4.96.4 Input Data Blocks

A, B, C, D, and E must be distinct matrices.

Note: Any of the input matrices may be purged.

#### 4.96.5 Output Data Blocks

X - Matrix.

The type of X is maximum of the types of A, B, C, D, E,  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $\delta$ ,  $\epsilon$ . The shape of X is the shape of A if A is present. Otherwise it is that of the first non-purged input.

Note: X cannot be purged.

#### 4.96.6 Parameters

ALPHA - Input-complex-no default value. This is the scalar multiplier for A.

BETA - Input-complex-no default value. This is the scalar multiplier for B.

GAMMA - Input-complex-no default value. This is the scalar multiplier for C.

DELTA - Input-complex-no default value. This is the scalar multiplier for D.

EPSLN - Input-complex-no default value. This is the scalar multiplier for E.

Note: If  $\text{Im}(\alpha)$ ,  $\text{Im}(\beta)$ ,  $\text{Im}(\gamma)$ ,  $\text{Im}(\delta)$  or  $\text{Im}(\epsilon) = 0.0$ , the parameter will be considered real.

#### 4.96.7 Method

If [A] is not purged, the number of columns, rows, and form of [X] = number of columns, rows, and form of [A]. Otherwise the descriptors of the first non-purged input are used. The type of

## MODULE FUNCTIONAL DESCRIPTIONS

[X] is the maximum compatible type of [A], [B], [C], [D], [E], ALPHA, BETA, GAMMA, DELTA and EPSLN. ALPHA, BETA, GAMMA, DELTA and EPSLN are assumed to be real if their imaginary parts are zero.

### 4.96.8 Subroutines

SADD - See subroutine description, Section 3.5.26.

### 4.96.9 Design Requirements

Open core is defined at /DADDA/.

### 4.96.10 Diagnostic Messages

None.



## FUNCTIONAL MODULE INPUT (INPUT GENERATOR)

### 4.97 FUNCTIONAL MODULE INPUT (INPUT GENERATOR)

#### 4.97.1 Entry Point

INPUT

#### 4.97.2 Purpose

Generates the bulk data input for a large number of academic test problems.

#### 4.97.3 DMAP Calling Sequence

INPUT I1,I2,I3,I4,I5 / 01,02,03,04,05 / C,N, $\alpha$  / C,N, $\beta$  / C,N, $\delta$  \$

#### 4.97.4 Input Data Blocks

Ii - As required by the execution of the module\*.

#### 4.97.5 Output Data Blocks

0i - As required by the execution of the module\*.

#### 4.97.6 Parameters

$\alpha$  - Problem Type Selector\* (Input, integer, default value = -1 (an illegal value for execution))

$\beta$  - Problem type option selector\* (Input, integer, default value = 0)

$\delta$  - Problem type option selector\* (Input, integer, default value = 0)

#### 4.97.7 Method

Based on the values of the parameters, INPUT reads, via FØRTRAN, one or more data cards from the input stream. Since the data deck has already been processed through the ENDDATA card at this point, these data cards always follow the ENDDATA card. Since FØRTRAN I/Ø is used, integer data on these cards must be right-justified. Once the data cards are read and checked, INPUT generates the table data blocks that would have been generated if the equivalent actual cards had appeared in the bulk data deck. These generated records are merged in with any coming from the corresponding input data block (generated by IFP) and are written onto the appropriate output data block.

## MODULE FUNCTIONAL DESCRIPTIONS

### 4.97.8 Subroutines

IUNION - Integer function which computes the union of constraint codes.

INPABD - Initializes the common block /INPUTA/

### 4.97.9 Design Requirements

Open core is defined at /INPUTX/ and must be sufficient to hold two GINØ buffers.

### 4.97.10 Diagnostic Messages

Many user messages are generated by INPUT. These are mostly related to improper or inconsistent data presented by the user and are usually self-explanatory. The messages generated internally within INPUT are 1738 through 1745. In addition, INPUT writes an echo of all data read from the input stream and certain informational output related to the processing that occurs while generating the user's problem data.

\*The workings of INPUT are described from the user's point of view in Section 2.6 of the NASTRAN User's Manual, NASA SP-222.

## FUNCTIONAL MODULE INPUTT1

### 4.98 FUNCTIONAL MODULE INPUTT1

#### 4.98.1 Entry Point

INPTT1

#### 4.98.2 Purpose

Recovers GINØ-written data blocks (tables or matrices) from User Tapes designated for that purpose (NASTRAN permanent GINØ files INPT, INP1, INP2, ---, and/or INP9). Normally, these tapes would be written by the companion module ØUTPUT1 (see Section 4.100) in a previous run.

#### 4.98.3 DMAP Calling Sequence

INPUTT1 / Ø1,Ø2,Ø3,Ø4,Ø5 / V,N,P1 / V,N,P2 / V,N,P3 \$

#### 4.98.4 Input Data Blocks

None.

#### 4.98.5 Output Data Blocks

Øi - Any data block which is to be recovered from the User Tape. Purged outputs (either implicit or explicit) are ignored.

#### 4.98.6 Parameters

P1 - Tape positioning option (Input, integer, default value = 0)

P2 - User Tape code (Input, integer, default value = 0)

P3 - User Tape Label (Input, alphanumeric, default value = 'XXXXXXXX')

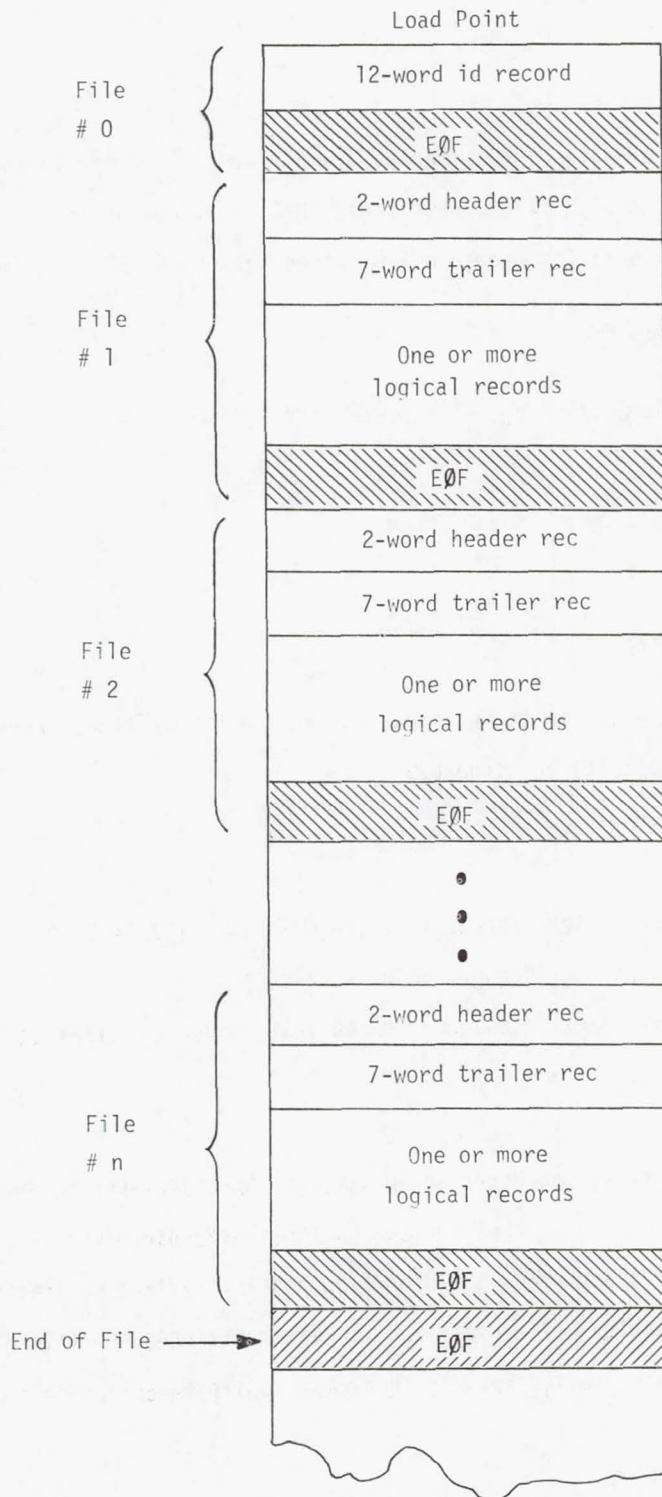
#### 4.98.7 Method

INPTT1 examines the first parameter and positions the User Tape designated by the second parameter (checking the User Tape Label defined by the third parameter if appropriate). INPTT1 then copies the next data blocks from the User Tape and writes them on the non-purged output data blocks in the DMAP instruction. The User Tape is left positioned wherever it is when the DMAP instruction requirements are satisfied. In this way, multiple calls can be made.



# MODULE FUNCTIONAL DESCRIPTIONS

The configuration of files and records on the User Tape is shown in the sketch below.



## FUNCTIONAL MODULE INPUT1

### 4.98.8 Subroutines

None.

### 4.98.9 Design Requirements

Open core is defined at /INP1XX/ and must be sufficient to hold two GINØ buffers plus one word of working core. Since blast I/Ø techniques are used, efficiency is enhanced by any additional core up to the longest logical record to be read in any one data block.

The User Tapes must be physical tapes.

### 4.98.10 Diagnostic Messages

Messages 3008, 4105, 4106, 4107, 4108, 4109, 4110, 4111, 4112, 4113, 4127, 4132, 4133, 4134, 4135, 4136, 4137, 4138, 4139, 4140, 4141, and 4142 may be issued.

In addition, when a file table of contents is requested, printout is generated giving the file number and the value of the first two words of the header record for each 'file' on the tape.

## FUNCTIONAL MODULE INPUTT2

### 4.99 FUNCTIONAL MODULE INPUTT2

#### 4.99.1 Entry Point:

INPTT2

#### 4.99.2 Purpose

Recovers FORTRAN-written data blocks (tables or matrices) from User Tapes. Any legitimate FORTRAN unit number not already utilized by NASTRAN may be used for this purpose. On the CDC machines, these unit numbers must also be compiled into the system in deck NASTRAN. Normally, these files would be written by the companion module OUTPUT2 (see Section 4.101) in a previous run. It is intended that files also be easily generated by external FORTRAN programs, however.

#### 4.99.3 DMAP Calling Sequence

INPUTT2 / 01,02,03,04,05 / V,N,P1 / V,N,P2 / V,N,P3 \$

#### 4.99.4 Input Data Blocks

None.

#### 4.99.5 Output Data Blocks

0i - Any data block which is to be recovered from the User Tape. Purged outputs (either implicit or explicit) are ignored.

#### 4.99.6 Parameters

P1 - File Positioning Option (Input, integer, default value = 0)

P2 - User Tape Code (Input, integer, default value = 11. This is the FORTRAN unit number for the file.

P3 - User Tape Label (Input, Alphanumeric, default value = 'XXXXXXXX'.)

#### 4.99.7 Method

INPTT2 examines the first parameter and positions the User Tape designed by the second parameter (checking the User Tape Label defined by the third parameter if appropriate). INPTT2 then copies the next data blocks from the User Tape and writes them (via GIN0) on the non-purged output data blocks in the DMAP instruction.



# MODULE FUNCTIONAL DESCRIPTIONS

The User Tape is left positioned wherever it is when the DMAP instruction requirements are satisfied. In this way, multiple calls may be made.

A description of the file configuration is given below for those FØRTRAN programmers who may wish to generate User Tapes with their own external programs for input to NASTRAN.

## Format of INPUT2/ØUTPUT2 File

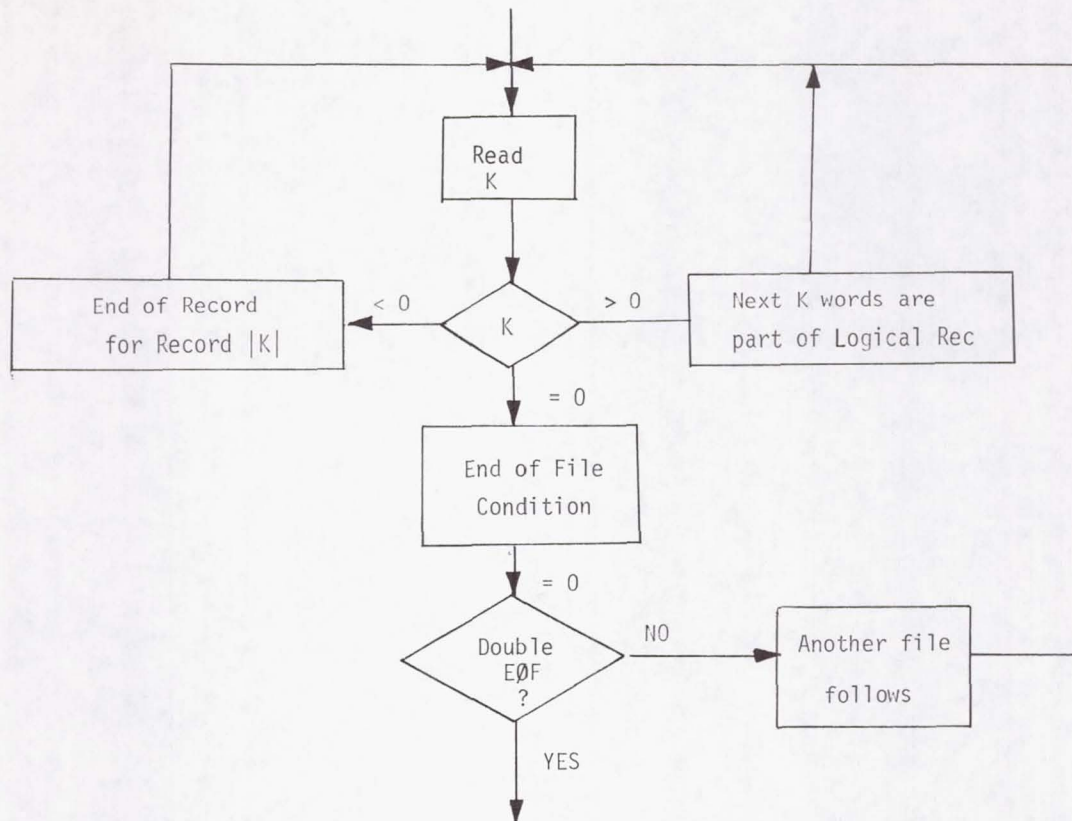
| NASTRAN File | Rec |                        | FØRTRAN Rec | Length |
|--------------|-----|------------------------|-------------|--------|
| 1            | 1   | KEY > 0                | 1           | 1      |
|              |     | { Data } ↔ KEY         | 2           | KEY    |
|              |     | KEY > 0                | 3           | 1      |
|              |     | { Data } ↔ KEY         | 4           | KEY    |
|              |     | KEY < 0      EØR       | 5           | 1      |
|              | 2   | KEY > 0                | 6           | 1      |
|              |     | { Data } ↔ KEY         | 7           | KEY    |
|              |     | KEY < 0      EØR       | 8           | 1      |
|              |     | KEY = 0      EØF       | 9           | 1      |
| 2            | 1   | KEY > 0                | 10          | 1      |
|              |     | { Data } ↔ KEY         | 11          | KEY    |
|              |     | KEY < 0      EØR       | 12          | 1      |
|              |     | KEY = 0      EØF       | 13          | 1      |
| 3            |     | KEY = 0      EØF = EØD | 14          | 1      |

### Restrictions:

1. Enough core must be available to hold the longest record segment.
2. A FØRTRAN unit must be available. On the CDC, this means that the PRØGRAM Deck (NASTRAN) must be re-compiled and Link 0 re-done if the value used for parameter P2 is other than 11.

## FUNCTIONAL MODULE INPUTT2

The logic by which the NASTRAN logical 'records' are interrogated is given in the sketch below:



### 4.99.8 Subroutines

None.

### 4.99.9 Design Requirements

Open core is defined at /INP2XX/ and must be sufficient to hold two GINØ buffers plus the longest FØRTRAN logical record on the User Tape. The 'User Tape' files may be on any FØRTRAN readable device.

### 4.99.10 Diagnostic Messages

Messages 2187, 2190, 3008, 4105, 4106, 4107, 4108, 4109, 4110, 4111, 4112, 4113, 4132, 4133, 4134, 4135, 4136, 4137, 4138, 4139, 4140, 4141, and 4142 may be issued.